Lee Hutchinson:

Hey folks, this is Lee Hutchinson with Ars. We are live on Zoom, and we're going to be getting into this thing in just a moment, probably another 90 seconds or so. So sit tight, and we will be going shortly.

Lee Hutchinson:

(silence)

Lee Hutchinson:

It's going to be just another minute, folks. I promise we'll get started soon. Keep on holding, and we will be right there.

Lee Hutchinson:

(silence)

Lee Hutchinson:

Just ironing out some minor last second technical issues. Almost there, sorry guys.

Lee Hutchinson:

(silence)

Lee Hutchinson:

I'm going to have to start telling jokes if this goes on much longer.

Lee Hutchinson:

(silence)

Lee Hutchinson:

For folks who are just joining, again, hey. This is Lee Hutchinson with Ars, apologizing for the technical issues we're having. We are almost there. I promise that we're almost there. We're just having the kind of tech issues that you have at the last second for everything, because that's just how life works because it's 2021. We're hoping to have this thing get squashed here in just a second. We are almost ready to start.

Lee Hutchinson:

(silence)

Lee Hutchinson:

I think we may have fixed our problem. All right, it looks like we've fixed our problem. Okay everybody, hi. Good afternoon or good morning, or whatever the time is where you happen to be, folks. Welcome to our special live chat event we're having on AI and machine learning. My name is Lee Hutchinson, and I'm the senior technology editor at Ars Technica. I'm joined here by my colleague and Ars info sec editor emeritus, Sean Gallagher.

Sean Gallagher:

Hey, Lee.

Lee Hutchinson:

And we also have with us Amazon senior principal technology evangelist Julien Simon, whose name I may or may not have just mangled. Hello, Julien.

Julien Simon:

That was pretty good, and hi everybody.

Lee Hutchinson:

Woo, all right.

Julien Simon:

Thanks for having me.

Lee Hutchinson:

Thank you. So the project we're going to talk about today here was my idea, so I'll take full responsibility for unleashing this upon the world here. But what we wanted to do was to create a machine learning model that could predict which out of a pair of Ars Technica front page headlines would win in an A/B test. So I naively thought this would involve feeding a model some previous headlines that won and some previous headlines that lost, and having the model do some natural language processing, and pick up on whatever commonalities there might be in winning headlines, and then Bob's your uncle, we're done.

Lee Hutchinson:

And reality proved considerably more complex. Not because this isn't a good machine learning task, but because as it turns out, even having a sample size that consists more or less of every headline we've ever written over the past five, six years didn't provide us with a big enough pile of data to work with I think. So we were able however to get a lot of neat work done here, and we learned a lot. And by we doing the work, I mean Sean doing the work with some assistance from Julien and friends. So Sean, is that a good tee-up? This is about where you came in.

Sean Gallagher:

Yeah, that's a succinct description. So we had about 5,500 headlines initially, or 5,500 pairs of headlines I should say, that-

Lee Hutchinson:

So 11,000-ish total strings of text?

Sean Gallagher:

11,000-ish total headlines from the cases where the headlines had a significant difference in the number of times they were clicked by people who viewed them. So we were able to assure that these headlines, there was something substantially different between the winners and losers. And then I tried, based on our original assumption of being able to pick a difference between winners and losers in terms of their content shows a couple different approaches to try and build a model to produce answers to tests of headlines. And that, I tried a couple different approaches initially. This is what we call the binary classification problem. It's either a winner or a loser. And unfortunately, based on the amount of data we had and also some of the issues with the data itself, we got just barely better than a coin flip in terms of accuracy. That was consistent across the board, and throughout what they call the confusion matrix of accuracy in machine learning. So it was equally getting things wrong on false negatives and false positives, and so it wasn't going to work as an accurate way of measuring anything.

Sean Gallagher:

And I took it to Julien and we probed a bit more, he took a few shots at it, and we didn't really get that much better. So what eventually happened was we had to bring out the big guns. Julien tried using a model called BERT, which is pre-trained on a large body of... there are several versions of it, but they're pre-trained on a large body of text. And that got a little bit better, and that's a warning sign right there. It was something that was just... if we couldn't get that much better with a bigger, pre-trained body of text to measure things, we weren't going to get anything out of it in the approach we were taking.

Sean Gallagher:

The final approach that we took was working with some folks at Amazon. I grabbed some other models based on BERT, one called SqueezeBERT, and another one that used the GPT-2 model, which comes from OpenAI, which is this huge model that we previously used to try to create deep fake news stories a few years back. Those got somewhat better results, and may have been something we could've gotten something production out of, but they were huge. SqueezeBERT is small. It got about a 60% accuracy rate, which is better than a coin flip, but it's still eh. But the larger model got up to close to 75% accuracy, but that larger model was a two gigabyte model, and would probably have required more compute horsepower than we would want to spend on checking headlines. That accurate-

Lee Hutchinson:

So let me jump in here. I have a question for Julien that I want to get out here before we fall too much further down here, because this is, as the dummy in the group, this is really interesting to me. Because I'm the manager, right? So this was all my idea, and I didn't have to do any of the work. So we started out with, Ars does A/B tests on headlines. We have all of our writers for every story that goes in put up two headlines. Both headlines are shown to varying samples of folks who visit the homepage. The headlines are shown for 10 minutes each, and we measure the number of folks who see the headline versus click on it, and that gives us a confidence interval between the two headlines. And then at the end of 10 minutes, whichever one is over our confidence interval, that's the headline we go with.

Lee Hutchinson:

And the data that we end up from that is, we have two pairs of headlines, the percentage of times they were clicked on, and then the confidence interval that was established of how the winning headline won. So Julien, my question for you here is sort of the same question that I asked when we started this crazy thing. Looking at this just from what I've described, what kind of ML problem is this really, and what would you do if you were going to solve this and it was a blank slate kind of thing?

Julien Simon:

Yeah, you're asking the most important question when it comes to machine learning. A lot of people obsess over algorithms and models, and as Sean explained, even getting the biggest gun in the arsenal didn't get to a very high quality model. And this is a good sign that probably the problem wasn't framed in the best possible way, and this is really the first step in every machine learning project. The business problem is very... I guess it's very straightforward when you just explain it, okay? You've got the headlines, you test. You A/B test them, you get some kind of metric, and then you decide which one is the best, okay? So it's very simple to explain. You don't have to be even a technical guy to figure it out.

Julien Simon:

Now, how do you frame this as a data driven problem that you can feed to an algorithm? And you can try different things. So initially we went for the simple thing first, which is what I really ask customers to try first. Try the simple things. Maybe they work, maybe they don't, but you get a baseline right? And that baseline tells you how much better you get when you try the fancier stuff, okay? And let's face it, in a lot of machine learning problems, the simple things work just fine. Basic linear regression, basic classification with algorithms like random forests or XGBoost. These are literally the bread and butter of machine learning practitioners.

Julien Simon:

So a lot of the time that's going to be just fine, and tweaking further with complex stuff is not going to work. So here in this case, the binary classification approach didn't work very well, even with the crazy complex models. So again, that's a sign we could try something else. And I think intuitively, intuition is very important in machine learning, we can see that maybe binary classification is a little naïve, because it breaks the link between the A title and the B title, right? Because in this case, we're not trying to figure out is this a good title or not. We're trying to figure out, here are two titles, which one of the two is the best? With respect to whatever metric you decide. So that relationship is lost when you try binary classification, or basically when you just treat titles in isolation.

Julien Simon:

So I think a more elaborate approach would be to consider those two titles together, okay? Treat that as maybe a single sequence, with some kind of quality score. And then say, okay, you look at this thing, and here's the score that's associated to it, and then go and learn that. In a sense, it's a little bit similar to fake news detection. There's a lot of work on that and you actually mentioned it. So here's the headline, the real thing, here's the fake news version of it. And sometimes, they're really similar. Sometimes it's just a few different words. And again, if you look at one or the other, both could be true. Both could be fake, right? But if you see both, then you see yeah, this one probably sounds more reasonable than the other version, right? And as a human, you kind of develop a sense for that, and then probably a model can develop a sense for that.

Julien Simon:

So I think if we went into a deeper project where we considered both pieces of text, both titles with a quality metric, certainly we would find that that works better, right? Because there's a relationship between the two, and [inaudible 00:17:39]. And the whole purpose of training is, let the algo figure out what that relationship is. Is it the number of words? Is it certain words you're using? We don't know, right? We don't know. And even if you ask domain experts, you've got your title experts as you explained earlier in the project, and they could say, "Well, I think this one's better." But if you ask them why, it's very difficult to explain why. It's a gut feeling, so that's what the model needs to pick up really. But I guess if you show them both and train them hard enough on probably a little more data than we have right now, I'm sure we'd get better results.

Lee Hutchinson:

Got it. And then-

Sean Gallagher:

I think that's fair.

Lee Hutchinson:

Sean, yeah. Sean, you ended up having to do a lot of work on this stuff anyway with the data prep and everything. Can you walk us through that a little bit?

Sean Gallagher:

Sure. So one of the concerns I had with data prep early on is that a lot of natural language processing tasks, you go through and you remove stop words and other things like that that would extract... are padding in a normal piece of text. But stop words are like 70% of headlines, right? So is punctuation, right? And punctuation makes a different in headlines. I mean, did I throw an em dash in there? Is there a colon? Those sort of things, they make a difference in terms of the impact of the headline, so I couldn't exclude those. The various models consume the text in different ways. So for things like BlazingText which we used, I needed to tag things in front for what they were, and it takes them in essentially a text flat file with tags on each line.

Sean Gallagher:

I was able to do all of this fairly simply with Python in a Jupyter notebook. And having spent a lot of time in Jupyter notebooks lately, and previous to this as well, it's become my preferred place to do things like this, manipulating large pieces of data, especially that comes out of comma separated values like what we've got for this test, because you can take the data structures and you can split them and merge them, and do all sorts of different things with the Pandas framework and other things. And PyTorch is sort of the standard right now for doing these sort of machine learning tasks now, which is based on Python, so it's really easy to move from one step to the other within the same environment.

Julien Simon:

Yeah, it's what everybody does. Experiment in Jupyter notebooks. There are other machine learning or data processing languages. I mean, some people use R, and that works very well for them. [inaudible 00:20:34], researchers, they love R. That's a very good one. But for better or worse, Python is kind of the main one right now. And Pandas-

Sean Gallagher:

[crosstalk 00:20:44] hated Python.

Julien Simon:

NumPy, scikit-learn has some good feature engineering libraries. So if people listening to this are confused and they want to figure out, okay, what should I learn first when it comes to machine learning? My advice would be brush up on Python, and you don't need to be an expert Python developer. I am not that by any means. I'm a mediocre Python developer, but I know enough to do what I have to do. And before you dive into ML, I would highly recommend that you check out like I said, NumPy, Pandas. It's the Swiss Army knife of data processing and data manipulation in general. And as we all know, that's 80% of your machine learning project right there.

Julien Simon:

So please be patient if you're just starting with this. Be patient. Don't dive into [inaudible 00:21:41] PyTorch and all those fancy libraries. You've got to do the homework, and the homework is basic Python, NumPy, Pandas, and a little bit of Jupyter, and then you are all set, okay? Now you've got the basic tools. And for experimentation and small scale stuff, it's fine. And then sure, if your projects get bigger and you need to scale that to big datasets, and there's again a production element to it, then you will need additional tools, additional infrastructure. But in the early days of the project, I mean you can work on your local machine, that's fine, right? Just use what you know, and-

Sean Gallagher:

Yeah, and I also-

Julien Simon:

Invest your time in those tools, and it will really pay dividends.

Sean Gallagher:

I also had to do a lot of initial work in text editors, because I mean, it was easier to do... I used a regular expression search to go through and find-

Julien Simon:

Yeah, regex, sure.

Sean Gallagher:

Things like HTML content. Had to kill all the HTML tags. All those EM tags and I-tags, and all the characters that were called out with an ampersand in front and all that had to be replaced with real text for processing, because we didn't want to have ampersand, amp, semicolon recognized as an entity.

Lee Hutchinson:

Some of the headlines ended up being... there's all kinds of corrupt in there. There's some tags, there's some HTML literals in there with smart quotes and dumb quotes. I wanted to interject just very, very briefly, folks who are on the Zoom participating, we would love questions too. I want to make this as interactive as we can, so if you guys have questions, just go ahead and type them in. And then when there is a break, we'll pull them up and start asking. I don't really have a set time as to when we're going to stop and do questions, but as they come in, we'll take them. So apologies for interrupting guys, [crosstalk 00:23:35] continue on.

Julien Simon:

No, no. Please ask all the questions. We are absolutely live, so you can ask all your questions.

Lee Hutchinson:

Yeah. I had one too, and it's one that I really wasn't clear on going into this, and this is going to sound kind of dumb in high level. But is there a good rule of thumb or rules of thumb for when you're approaching an NLP problem like this, a natural language processing problem like this, is there a good rule of thumb for which off the shelf model to kind of pick to start with? And how do you figure that?

Julien Simon:

Yeah, it's a good, again it's a good one. And I would say it depends on how much data you have, right? First of all. NLP means most of the time is going to mean deep learning, and deep learning generally requires quite a bit of data to get good performance okay, compared to I would say traditional statistical algorithms. So if you had 10 million headlines, okay? You could say, "I can probably train from scratch." So I can probably gather one of those reasonably fancy algos like BERT or a variant of BERT, and I can probably train it completely from scratch and specialize it on this particular problem, okay? And this goes, this works well. So if you have let's say very specific vocabulary, like let's say you want to train a model specifically on let's say legal documents, or life sciences documents okay, or medical documents. And God knows there is quite a bit of jargon there, right? We complain about IT jargon, but both my parents are medical doctors, and there's a fair bit of jargon there.

Julien Simon:

So you would say, "Well, a general purpose model, that probably won't work, so let's go and train something custom on my own dataset." But a lot of time, you will not have that amount of data, right? You will have let's say thousands or tens of thousands of documents. It's unlikely you're going to have 10 million legal contracts or 10 million patient notes, okay? Yeah, those datasets are out there, but most of the time you don't have that much. So in that case, you need to use pre-trained models, okay? And use a technique called fine tuning, where you actually take... you start from an off the shelf model, so a model that was already trained on a very, very large corpus. Typically Wikipedia and that kind of thing, right? Big datasets, bulky ones.

Julien Simon:

And so they've already learned how to extract patterns, let's say in English language, okay? They already know about English language, and that word next to that word means this, et cetera, et cetera, right? And so you can then fine tune it on whatever data you have, and let the model figure out these extra domain specific patterns that you have in your dataset, okay? And that's generally very fast. Obviously fast means, fast to train means much less expensive, and you don't have to wait for three days for your model to train, and you can iterate faster. So that's really the first thing. How much data do you have, do you want to train from scratch, do you want to fine tune, okay?

Julien Simon:

And then which variant of BERT, because BERT tends to be the main one these days, but again there are other ones. You have to do some research. You have to see, you have to find problems that are kind of similar to yours, go and read technical blog posts, go and read research papers, spend some time researching, do the homework once again, and see what works best for things that look like your problem, okay? And that's always a good starting point, okay? It's like going on YouTube and looking for a tutorial to do whatever you want to do in house, right? So the same works for ML, except you probably won't find the answer on YouTube. You'll find the answer in technical blog posts and research papers. But yeah, doing the homework again pays dividends, because you will focus on two, three, four likely techniques that should work and you can take it from there.

Julien Simon:

It's safe to assume your problem has already been tackled somewhere or something close to it, so don't go and reinvent the wheel. Be super pragmatic about it, and then try, again try the simple things first. Yes, please try the basic algos and then go up in complexity. And yeah, and just iterate and go as far as you can.

Sean Gallagher:

So we definitely had some prior cases that I looked at in research that had looked at headlines, but they were looking at much vaster collections of headlines. So there was a click bait study I looked at that-

Julien Simon:

Yeah, click bait. Sure.

Sean Gallagher:

Yeah, that went in, which is very similar to what we were trying to do here, but it was looking at headlines from a particular online news source and comparing it to headlines from established news sources, and trying to determine what click bait was and things like that. And that was trained on a much, much larger corpus of headlines than what we had available to work with here. So context is the most important thing in the dataset for this, just making sure that you have enough context for what the content is to be able to make a judgment on it. And that's why one of the things I said to Lee when we got rolling was, "Wow, I wish we had done this with comments instead of headlines," because the [crosstalk 00:29:40]-

Julien Simon:

Sure, like product descriptions. Customer reviews. There's a really cool dataset which is called the Amazon Reviews dataset, it's actually hosted on AWS. So if you look for it you'll find it, and it's humongous. It's tens of millions, maybe hundreds of millions of reviews, so pick your poison. If you want to build a model for video games, you've got as much as you want in there. And yeah, it's huge, and obviously you can build very fancy apps with that because there is so much data to pick from. Another interesting point that we haven't brought up, and I remember we had this discussion early on, was what about words that could be so powerful like company names or billionaire names, famous people, right?

Sean Gallagher:

Certain president names, [crosstalk 00:30:37].

Julien Simon:

Exactly. And should we hide them or not, right? Remember, we said should we replace all company names with "company?" Should we replace all CEO names and investor names with "person" or whatever? Because obviously... and I guess it's fair, let's take Elon Musk as an example. If you see a title with Elon Musk, you see Elon Musk and you're going to click, right? I'm not saying everybody will, but it's like, "Oh, what's up with him now?"

Sean Gallagher:

Most people do. Most people do.

Julien Simon:

Yeah, what's up with him? He's doing all kinds of interesting, and yeah, interesting things. Sometimes a little bizarre too, but yeah, he's generally a very interesting character.

Sean Gallagher:

Yeah, and I tried that a little bit too early on. I did go through and try to do substitution for person and company and things like that, and I didn't get anything out of it really. And part of it is because we were already canceling out company names and person names, because we had the company name, person name, in both headlines usually. So it wasn't really an indicating of preference for one or the other. I mean, I did do a little bit of a look at, prior to us doing this, did a word cloud on what words were more prevalent in winning headlines than losing headlines, just to get a sense of if there was a real issue. And the words that popped up in the losing headlines more frequently than the winning headlines were not names or companies, they were [crosstalk 00:32:17]-

Julien Simon:

Yeah, so that's a really interesting thing. That kind of analysis is super important. And again, it's not machine learning per se. It's really data wrangling, data analysis. I mean, as a writer, you could be totally convinced that if you put Amazon in a title, or if you put Elon Musk, it's just going to click better, because maybe you had a few instances of that in your own experience. And that could be a very strong belief, but maybe that belief doesn't actually survive deep analysis. So that's why all those... intuition is important, gut feeling is important, but you've got to back it with data, right? Because somebody could say, "Oh right, yeah. We're going to over-index on billionaire names and company names, because we know that clicks better." And well, maybe, maybe not. And in this case, it looks like it's actually not the key factor, right? It's more subtle than that. But that's a really good, interesting-

Sean Gallagher:

It doesn't help [crosstalk 00:33:26].

Julien Simon:

That's a very interesting learning, I think.

Sean Gallagher:

Yeah. It doesn't distinguish between a winning headline and a losing headline on the same topic, and that's the problem. And so because both the headlines are going to mention the subject, and they both would probably be successful based on the topic regardless, but one is more successful in drawing clicks than another for some arbitrary reason, and that's often because of information that's in the headline that is contained in words that are not subject related. So for example, headlines that were a little bit less definitive that contained weasel words like "may" or things like that, they tended to do more poorly than headlines that were very specific and actionable, and more not in passive voice. Not couched. So that wasn't a real surprise to me having written a lot of headlines in my day, but it did definitely show that the content of the headline beyond the nouns and the action verbs mattered, because they determined what... the preferences of the readers were towards more active headlines.

Lee Hutchinson:

Let me jump in real quick here, guys. We've had a couple of questions show up, and the first one was from someone asking if we could sort of go over the tools that were used. So rather than do that, A they'll show up in the transcription for this, which will be available on arstechnica.com, and also Sean's article series has all of these tools mentioned in them. So the model names, BlazingText and Hugging Face and BERT, and all of the tools that we used are in the piece, and then they will also be on the recap that gets posted with this.

Lee Hutchinson:

An additional question though, and this is maybe or maybe not germane. The question is, can you clarify... this is partially for me. Can you clarify what you stated about the original A/B test, and wouldn't the alteration of headline presentation too soon affect the data used in the analysis? So I'm not a statistician and I don't have a good math kind of answer to that, but I can tell you that the way the A/B testing works on Ars, which is germane to this discussion. So each writer generates a pair of headlines finally. We actually have a bunch that get workshopped. And then each writer comes to the table with a pair of headlines that are attached to the article. Both headlines are shown to a subset of visitors. So some, as soon as the story gets published, some folks see one headline and some folks see the other. It's randomly determined.

Lee Hutchinson:

Now we do pick... the A headline is used to set up the permalink for the article, and to do the tweeting and stuff, because all that has to have a URL assigned to it. So the A headline ends up becoming the permalink. But then the A/B test runs for 10 minutes, and it runs for that 10 minutes for every article. So whether shortening or lengthening that time affects the analysis of everything, I don't know if there's an answer to that. But I do know that all the articles get the exact same thing, and then the confidence interval that we established is 90%. So if we don't have an established 90% confidence interval between that that A/B test ends up with, then we throw away the B headline and just go with the A. Rather than trying to do then some sort of complicated secondary analysis that may or may not be useful, we just go with the A.

Lee Hutchinson:

So the idea here was to see if there's a way to nudge us in a direction towards writing headlines that... this would be one tool in the headline writing toolbox, but see if we could get nudged in the direction of writing headlines that are more likely to win. And this wouldn't be like embracing click bait, because that's a whole other side discussion, right? And that's a very complicated one, and one that I know we've gone down before. And like you said, there've been AI/ML experiments done before with click bait headlines and stuff. There's datasets that have been established out of that that are available to be looked at. We really just wanted to sort of look and see, what were the commonalities between headlines that won?

Lee Hutchinson:

There was another really good question that builds on this, and the question is, is there a way to quantify the human, gut feeling of a winning headline or not? And that's a really good question, and I don't... I mean, that's really a question for Julien more than anything else. I mean Sean, I'll go to you first, and then I'll kick this to Julien. So you've said you've written headlines for quite some time, I know you have because I've edited many of your pieces and seen many of your headlines. It's a skill that you sort of develop. So the question is, is there a way to quantify the human gut feeling of whether a headline is good or not, and is there a way to quantify it in ML terms? And I don't know if there is.

Sean Gallagher:

Well, I think there's a grammatical structure that goes with a good headline. I mean, you've got to... the less obtuse your headline is about what you're saying the better. I mean, you don't always get a win out of a direct headline, right? As Beth Mole demonstrates on a regular basis. But there's got to be something that's got a hook to the headline that draws interest, and it has to be something immediate, because you only have about 80 characters to do it in. So I've been right on my headline choices based on the A/B test about, I don't know, 70% of the time. Sometimes I'm disappointed by what the crowd chooses. But that's life. But still, I mean, so if I'm picking right about 70%, I'm doing about as well as the two gigabyte, trained on every piece of text known to man headline out there.

Julien Simon:

That thing, yeah. It's the incompressible error.

Sean Gallagher:

Yeah.

Julien Simon:

So exactly. It's a machine learning concept, and it's an important one. When you're trying to shoot for, trying to set expectations on how well could ML be doing at this, one thing you look is how well are humans doing at this, okay? So if it's about classifying animal pictures, so you could take a panel of 100 people and show them animal pictures, and give them a limited time to call out, okay, whatever species this is, and you measure error. Okay?

Julien Simon:

And you'll find that there's no such thing as 100% accuracy even for humans. Especially for humans I want to say. And it's the incompressible error. So it's, you will never ever solve that problem. Not with humans and not with machine learning models. So coming back to Lee's original point, I think if I was spending a lot of time on this, I would talk to Sean and I would talk to you, I would talk to writers, I would talk to people that are considered very good title writers, okay? So we would sit down with a cup of coffee, and I would try to drill holes in their brain and make them tell me as precisely as possible what makes a good title. So Sean actually gave us a good hint, he said complexity. So you could think AI is magical and say, "Oh, my two gig model is going to pick up whatever complexity is in the sentence." And okay, all right. Maybe, maybe not. But we could say, well so the text is obviously important, but what if we added another feature right? What if we added another feature into the mix?

Julien Simon:

And we could have a complexity metric. I mean, that stuff exists for English and I'm guessing other languages, right? You have tools that give you a numerical score, a readability score, whatever you want to call it. And you could actually use this in pre-processing step. Compute that score and add it as a feature, okay? And you could add, I don't know, word count, and you could add the number of words that have more than, I don't know, 12 characters. Or what's the level of speech in there? Are you using very fancy verbs? Sometimes that really confuses me. I mean, people want to, I guess they want to sound really clever, but with my broken English I do stumble on the words or verbs that sound very literate. And I'm like, "What are these? What does this mean?" That doesn't make me want to click, because I have no idea what that article is about.

Julien Simon:

So those could be good tracks to explore. And talking to a bunch of people, a few things like this would surface, and they would be fuzzy and they would be... in my experience, I think that or... and it's, okay, how do we make that a little more machine learning-y? Can we put a metric on that? We need a number. And we could add those as features and explore the problem, and see if those metrics are actually important. And it's that discovery problem, which is really framing the problem and discovering it, and then evolving the framing and exploring some more. That iterative phase at the very early stages of the project is super important. And again, don't rush to tooling, and don't rush to complex stuff, because garbage in, garbage out, right? It was true in '70s, it's still true with ML.

Julien Simon:

So try to be creative, spend time with your domain experts. Be nice to them because they know, and they need to share that expertise with you in the most useful way. So then it's your job to say, "Okay, I'm going to sum up this gut feeling into a metric that I can compute and add that to my model, to my feature, to my dataset," and see if the model picks up some kind of relationship. That's what machine learning is about, exploring.

Lee Hutchinson:

Absolutely. So we are unfortunately out of time. We need to kind of wrap here very quickly. So Sean, I wanted to kick this to you kind of at the end here. If we were going to do this again, can you give me the 30 second version of where we'd start differently, and then bring us to close here?

Sean Gallagher:

Okay. Well we would start off by having a little more thought about what our actual business problem was here, and what the problem we were trying to solve was. I think one of the big problems we had was that we came in with an assumption that a headline by itself carried enough information for us to make a judgment on, and that clearly wasn't the right path. So we can't just test one headline to see if it's going to be a good headline without the context of what are the other options. So you need to do that, and so you need to have the context of both headlines. We should've been testing both headlines at the same time, which we did later with more success. And honestly if I had to do this again, I would probably choose a different problem.

Julien Simon:

Yeah, it's not an easy problem. Like I said, that sentence comparison thing, and it could be click bait, fake news. It has a lot of variations. It's not easy because it's language, and language is very subtle, right? And the relationship and how people react to certain words is very subtle. So you can try and brute force the problem, which worked up to a point, but if you need to go and brute force it, problem yeah, you should go back to the whiteboard and say, "Okay, let's take a different angle. Let's try to solve this in a different way."

Julien Simon:

And again, it's perfectly okay to do that. It's not failure, it's experimenting, learning. And along the way, you actually understand the problem much better now than you originally did, right? And you just need to go, again, go back to the whiteboard and try to frame this a little bit different, and go back to working with the data and models, and you get closer every time, right?

Lee Hutchinson:

Indeed, indeed.

Julien Simon:

That's what real life machine learning is, iterating.

Lee Hutchinson:

Absolutely. Well thank you. Thank you Julien, your expertise has been appreciated. Thank you for helping out with this project, this has been exciting and a learning experience.

Julien Simon:

Thanks.

Lee Hutchinson:

Sean, thank you also very much. Thank you for being in the trenches on this one, and also appreciate everyone showing up for this event today. Thank you guys very much. The transcript will be posted online.

Julien Simon:

All right, thank you everybody. Bye-bye.

Sean Gallagher:

Thanks, Lee.

Lee Hutchinson:

Thank you everybody.