Okay, next up, we have up enough. So I've been up is a high schooler at the Jefferson High School in Virginia. And he will be talking about using Transformers in work, which is actually the same technology that Barack was talking about, but different in an entirely different context. He's going to be talking about it, but Twitter feeds. Thank you. Go ahead and get started.

Thank you. Oh, sorry. Okay, is everyone able to see the presentation? Clearly? Yes, you're good.

So yeah, my name is Ivan od. And the project I did was utilizing Bert for natural disaster, for reading tweets, and from these tweets, determining whether that tweet subject matter was linked to natural disasters or not. And so the reason for this was pretty simple, right? reason for this was pretty simple, because natural disasters are incredibly devastating, right. And so by reading social media feeds, and using them to determine whether, you know, the
specific post is talking about a natural disaster, an earthquake, a tsunami, a snowstorm, whatever, that lets us essentially create a sort of warning system, right? We use our model on live data, we can we can apply this model right to live Twitter data. And then if the majority of these tweets seem to be talking about natural disasters, then we can use our warning system and be like, yes, there's likely a natural disaster happening right now. Now, the second motivation for this project has to do with Bert itself, right? So Bert is a very recent development in natural language processing. But despite that, it's already incredibly impactful. It's becoming currently notorious in the field of natural language processing. So Bert stands for bi directional encoder representation for bi directional encoder representations from transformers. That is what the full acronym means. And there's not enough time to explain it fully. But throughout the presentation, I'll try to give everyone a general idea of how the architecture works. And so essentially, Bert is incredibly powerful. And with this project, I was hoping to develop a proof of concept, right? If it's true, that bird is able to detect whether tweets are talking about natural disasters or not, then we can extend this to other applications, right, involving social media, or whatever. So this specific solution I use, right? So I did a multitude of experiments. But the final experiment, the final model, worked with this with this specific pipeline. So the pipeline was as follows, right? First, Bert was utilized for feature extraction purposes. And then from there, we passed those features into a convolutional neural network as embeddings. And by doing this, by doing this, we were able to classify each tweet into one of those two distinct categories. And so this method had incredibly high accuracy. See, and you'll find out about that in a later slide. So the data, so the data used, right? So the initial data set was simply just a kaggle data set, right? But then from there, I took that data set and cleaned and curated it, I dropped null values I got rid of. I got rid of I cleaned the tags, I got rid of all the ones that seemed weird. And, you know, so with the text, what I did was I got rid of stop words, I got rid of unnecessary characters and symbols, right? Because they just won't tell us anything. Right? Those tell us that we ready for disaster. And then once I did all that the number of disasters between from a

data set was 3004 71. And the number of disaster unrelated tweets in a data set was 4342. And you can look right here, and you can see that, you can see some details about the data set. You can see how most of the tweets are on the longer side, right, because you can see how there's like the spike. And in addition, you can see that the tweets that were related to disasters, were more likely to be shorter than the ones unrelated disasters, if that makes sense. And then right here at the bottom, you can see, you can just see the actual tweets and can see the text clean. So the text is the original tweets. The target is the classification, whether it's really the disasters or not. And the text clean is the you know, the new text once I got rid of the stop words and special characters and whatnot. So the actual AI, so basically just explain how Bert works, right? So Bert is a semi supervised learning model, which means it takes advantage of both unsupervised and supervised learning techniques. All right. And so when people are referring to using Bert for a product jack, what they mean? Is there a specific Bert model because there's actually a bunch of different Bert models. Now the two main architecture is when they're talking when people talk about Bert, our Bert bass and Bert large, the only real difference between Bert bass and Bert large has to do with their size. Right? So Bert bass has only 12 layers of encoder model and Bert large consists of twice as many with 24 layers and encoder model. Also, Bert bass has 768 hidden units, while Bert large has 1024 hidden units. Also, Bert bass has 110 million parameters and Bert large history of 40 million parameters. And so because of this, Bert large is much more powerful than bird base, but also because of how much heavier and larger it is.

It's not always practical to implement, right? Because it's just so much slower, right? The inference type of bird large is comparatively higher than that of bird base. All right, so in order to actually create this model, this program, what we what I did was I use TensorFlow and Python. And that's, that's the general gist of it. Right? It was a very, there's no fancy technology used beyond this, just Python and in the TensorFlow library, and you know, other related machine learning libraries like SK learn and all that other stuff. And so that was a very straightforward process. There's nothing too fancy about discussing the program, the actual technology. Now on to the actual experimentation, right. So the first experiment, it was a very straightforward pipeline, I used a count vectorizer, then I use a TF IDF transformer. And then from there, I just used a simple multi, multi gnomeo Naive Bayes model. There isn't enough time in today's presentation to describe what all those terms mean. But just as a quick summary, count, vectorizer is just a way of tokenization TF IDF is a way of getting a specific representation. And then from there, the multi node multinomial Naive Bayes model is just a probabilistic learning method that's really popular natural language processing. The second approach, experimentation was just logistic regression, there's no fancy pipeline here, all I did was I threw a logistic regression model, a sigmoid function at the data. And that was it. And yet, despite it being so much simpler than the first method, it got almost identical results just a little smaller. Now, the final approach was the was the one that we actually care about that as a final experimentation. And so essentially, how this one works is we first we removed all the punctuation stopwords. Right to get the clean text dropped in on

values, and all that, then we did was we use something called Muse diverts full tokenizer. And what that did was, what that did was, it helped tokenize the data, right, which is important. And then from there, we use the Bert base model for feature extraction. And then from there, we use these Bert model features as the embeddings to our fine tuned convolutional neural network model. Now, you can see right here in the figure what that convolutional neural network model looks like, right, but just to go into a little bit more detail, the way it worked was there was an embedding layer. And then there were three pairs of convolution, one dimensional layers, right, each of which use the relu activation function, and then one dimensional max pooling layers. And afterwards, they just had three pairs of layers that are concat. And these three pairs of letters are concatenated, pass through a dense Rayleigh layer, a drop out layer, and then finally, a dense sigmoid layer. If you don't understand some of that terminology, that's okay, it's not really important. All you need to understand is just, there's a bunch of different layers. And there's a bunch of different math involved with those layers. Beyond that, it's not too important to understand the details for the sake of this presentation. And so just some learnings from this presentation. burts, powerful, very straightforward. It's true, the final approach was wasn't pure Bert, right. It was also, you know, Sienna, and there's a lot of other stuff involved. So to say that the only reason the models successful was because a bird would be unfair to the other developers machine learning that made it possible. But if you consider all the awesome things bird has done over the past three years or so, it's very obvious that bird had played a big role in this model success, right? That model we created that pipeline we created was incredibly impressive, right? It got near 100% accuracy. 99.73% accuracy is very high, right? Having accuracy for this type of model, right? Anywhere in the 90s would be particularly impressive, but 99 point 73% accuracy is just insane.
Alright, so for the future. I'd like to flesh this project out more static VA more proper warning system for natural disasters, maybe make it a proper app or website. Because essentially, the goal for this product is to make a system for people who don't keep up to date with the news and whatnot, right? These people might not be typically aware of these sorts of things, natural disasters and all that and so this would be a good early warning system. And now you might be thinking right there already exists warning systems for natural disasters. But the idea here is that by using social media posts from other areas that we might be able to find out about disaster before becomes worth mentioning on the news. Now, of course, this sounds like a bit of a stretch of the imagination, right? Nowadays, whenever there's a big storm or something like that the news will cover weeks, even months in advance, right? Because they're always monitoring the weather. That's their job. But who knows, this could be like an extra tool for the more unexpected ones. Now, again, you might be skeptical, right? Isn't it?

Kind of naive to think that of a natural disaster so sudden that the news doesn't know about it? That some dude on Twitter with right, it doesn't make sense? And yes, there are admittedly lots of flaws in this idea. But the point is that this project was basically a basic test of a proof of concept, right? And now that we know that this sort of thing is possible, that bird is this powerful, we can flesh it out and maybe use other types of data sources, we can build something really incredible. It doesn't have to be whether natural disasters related stuff necessarily, but just the fact that this is possible at all, it is possible to predict this sort of thing with this degree of accuracy and power.

But that alone means a lot for future endeavors. So yeah, thank you. Any questions? Thank you very much. I think you may have answered a lot of the questions. I think I just wanted to kind of add one thing, which is, I think we're seeing particularly when as we look in the news that you know, there are many countries in the world where communication is either, you know, structurally impossible, or is being blocked. And in many of those cases, some of the institutional structures don't work. And you know, social media has clearly been seen as an alternative way to communicate. So I think there's many, you know, options like that. So thank you very much.