Extracting names and other entities from text

We've done a lot of work with images, and that's great for learning what machine learning can and cannot do. But as journalists, a lot of the problems that we run into surround text documents, a trove of documents you might get from a source or records request. So we're gonna look at how we might use machine learning to go through some of those big document dumps.

The first thing we're gonna look at is entity extraction. Entities are things in documents like people's names, company names, city names. And we don't know what to search for, we don't know maybe what names are in there, but we just want to see all the names that can be identified and a machine learning model can help with that. There are a couple of ways to do this out in the world. Google has a couple of services that do this. And even with document cloud at documentcloud.org, you can upload PDFs and extract entities, but I'm going to show you how to do this right in your own notebook and get a sense of what's possible.

Let's go back to colab and as usual, go to Github, "quartz". This time we're going to go to ff-extracting-entities-with-spacy. We're going to actually look through a pile of documents in this case, e-mails that were part of a court case and look for people's names. Names are one of several entities that you can search for using a pre trained model called spaCy. We're gonna use the large English language spacy model to extract those entities from a large pile of emails. So we definitely are going to need to change our runtime to GPU, and so we remember to do that.

And everybody who's doing this notebook can run this cell right here. And this is going to load the big spacy model, which is many megabytes and the spacy software, which is also pretty bulky, and so this could take a few minutes. In fact, it can take up to about three minutes when I clocked it. OK. A few minutes later and we're all set. There's a lot of text that came out, but this is the key part here. "Download and installation successful". So we're good to go. Next, we're going to get the data. These are the e-mails that we're going to search through. This actually was a data dump of 4000 New York City e-mails from the mayor's office. And we're only going to look at the first hundred pages for this exercise, so let's go ahead and download that. And that's done. And we can just look and see, I like to look and see what we have. There it is, it's a PDF. And we got it stored right there in the data file. All right. So let's try entity extraction.

So here what we're gonna do is we're going to load that whole English language, large model into this, "nlp" variable. So this is a pre trained model. It's been trained on all sorts of information, including broadcast news and telephone conversations and whole bunches of things. And now we can give it a try. So we're going to say that the document we're just going to test this out. So set aside the emails for a second. The document is going to be this phrase "San Francisco considers banning sidewalk delivery robots". OK. That's our document. And then what we can do is that we can say, go through and tell us all of the entities that you find in that document. That's what this is. So it's going to go through all the document entities and we're going to print out every one of them. Well, there's only one, it's San Francisco, and it's a GPE, and that stands for counties, cities, states, that's the description of GPE.

Actually, you can see the whole list of entities here. It can find people, nationalities, facilities, organizations, so GPE is countries, cities and states. So products, events, works of art, it will find all of those kinds of things inside of the text. So here's another one, here's a little story I put in. "John drove his Volkswagen Golf," that's a car, "north on Interstate 35," that's a highway "to Duluth, Minnesota," which is a city in the United States "where he stopped at the Aerial Lift Bridge and looked out over Lake Superior." So if I say that that is my story and I make the document the "nlp" of my story, I can actually go up here back to this cell so and run this again with my new document. And it says, "John is a PERSON" and that's people, including fictional people, the Volkswagen Golf, it says it's an organization, and that's interesting. That's actually a car, but it's picking up on Volkswagen. So companies, agencies, institutions, etcetera. Interstate 35, that's a facility, in fact, it's a highway. Duluth is a city or a state, it's a city. Minnesota is a state. And the Aerial Lift Bridge is the facility, it's actually a bridge. So you can see how it is pulling out these entities. That's pretty, pretty cool.

Ok, so the next thing we're going to do is we're going to look at those e-mails. This block of text you got this block of code, you don't have to really know what it's all doing. You can look through it if you like. And if you understand Python, it'll make even more sense. But basically what it's doing is it's loading the PDF into a data format called "jsonl" and jsonl just puts each page of a PDF on a single line of data, so let's take a look at the beginning of that file. That's what the head of the file is. And you can see here's each line and there's something called source and content, and that is the content of the e-mail. You can actually see the different texts of the e-mails and actually toward the ending of them, each one of them has a I.D., and that's really the page number that that or the line number that assigned. So here's page three.

So, we have content, we have-- all you need to know is we have content of the emails and which page of the PDF it's on. Ok, so let's find and list all of the names that are in that now jsonl file. So we're going to do is we're going to open the jsonl file. We're going to go through every line of the jsonl file. That's the loop here for a line in the file F, and then we're going to do, we're just going to read that in to this variable and we're going to get the text from that line. That was the source content. We're going to get the page number from that line. That was the I.D. And then we're going to run the "nlp" on that. "NLP" means Natural Language Processing, we're going to run natural language processing on the text, the text of that line. That'll be that page. And then we're gonna do our same little loop again. We're say, look for all the entities and loop through them, and then if it's a person, then print out the page number in the text of that entity. Ok. Ready? Here we go. I'll show you what that does.

So basically what it's doing is it's going through every page and it's finding every name on the page. So on page one, it has two names, jonathan Rosen, Jimmy Pan. On page three apparently is Phil. So that's sort of useful, except if we're going to look through if somebody whose name appeared on many pages, that's a little bit hard. It's a little bit better, but this code actually, and I won't go through every line here, but this actually organizes them a little bit nicer. And in fact, if I list this, you can see. Ok, so here now we have in alphabetical order. We have the names of the people and the pages they appear on. So if you can see here, Alison Bauman, she appears, looks like she appears five times on page 71 and three times on page 78 and on page 80. There's actually a little bit nicer way to do this. This will actually make it even clearer.

And now we can really start to look through and say, oh ok, well, here's 100 pages of documents. I can just sort of scroll through here and see what kind of names I might recognize. So Bill de Blasio, is the mayor. Clearly misspelled or misread, scanned maybe as Bill de "Blasia." So, you know, that's something that you'd have to consider, and there's a bunch of different names. You might just go through and try to see which ones as a journalist you recognize. I'll tell you, I recognized Gwyneth Paltrow. So Gwyneth Paltrow is an American actress and appears in these e-mails. Well, that's interesting. I never would have thought to search the city e-mails for Gwyneth Paltrow. So, I mean, what? Why did she appear in these e-mails? Well, at least her name did. And so if I click on the original document with the link at the bottom of the page, I can just scroll to page three or, you know, I could because I am looking for her, I could just search for it and find, oh there is her name. But I wouldn't have known to search before spacy had taken out all the names, right? So now I can see that, oh, it looks like somebody was asking about Gwyneth Paltrow in this e-mail.

[00:11:08] Next, we'll learn how to take a pile of documents and sort them into two piles based on their content.