THEORY OF COMPUTATION (CSE2002)

NAME	ANITEJ SRIVASTAVA
REGISTRATION NUMBER	19BCE0835
SEMESTER	WINSEM 20-21
COURSE	CSE2002
FACULTY	Dr. Viswanathan P
SLOT	A2
ASSIGNMENT NUMBER	1
NO. OF MEMBERS	1

GOOGLE COLAB FILE LINK:

https://colab.research.google.com/drive/1hIkZl6yTsHPdA3l-xDo8ZquDjpEoePD8?usp=sharing

Directed Acyclic Graph Construction for Malware Detection/Analysis in Cybersecurity & Digital Forensics Domain

Step 1: Acquiring raw text with respect to malware attacks & threats in the cybersecurity domain from AlienVault - Open Threat Exchange

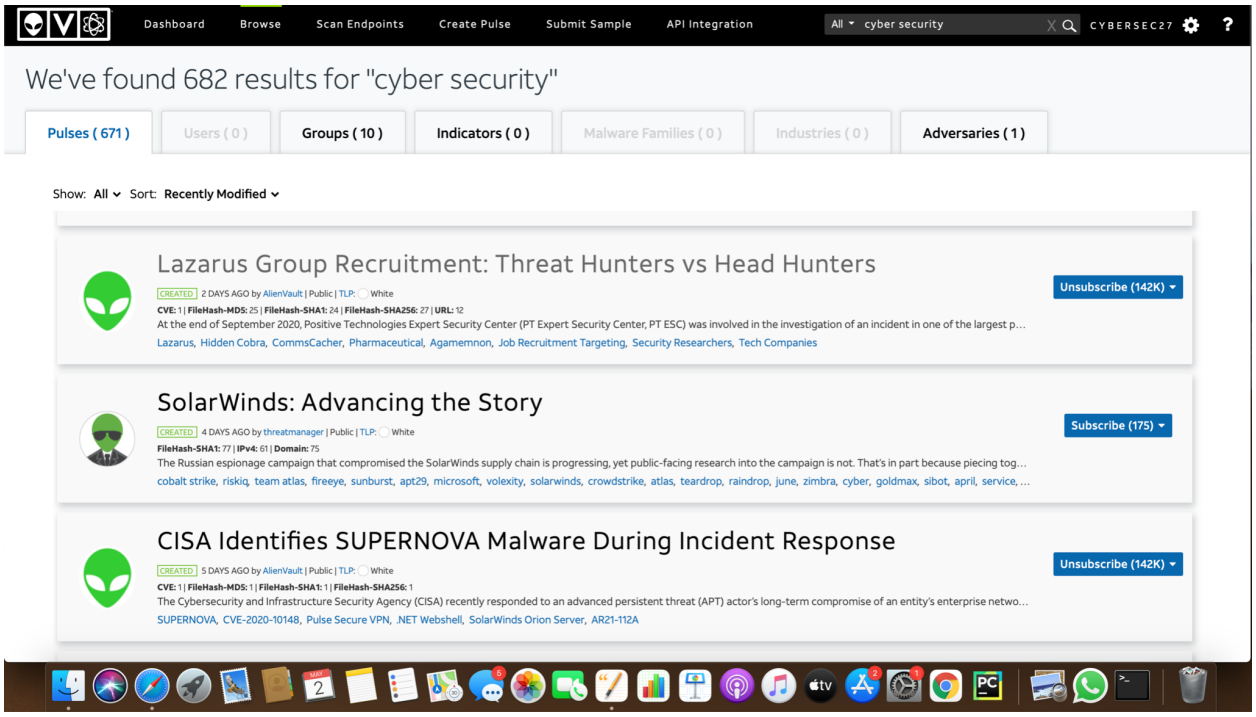


Figure 1: AlienVault - OTX Database

Collected the data by scraping text from all the pulse reference links on the OTX platform.

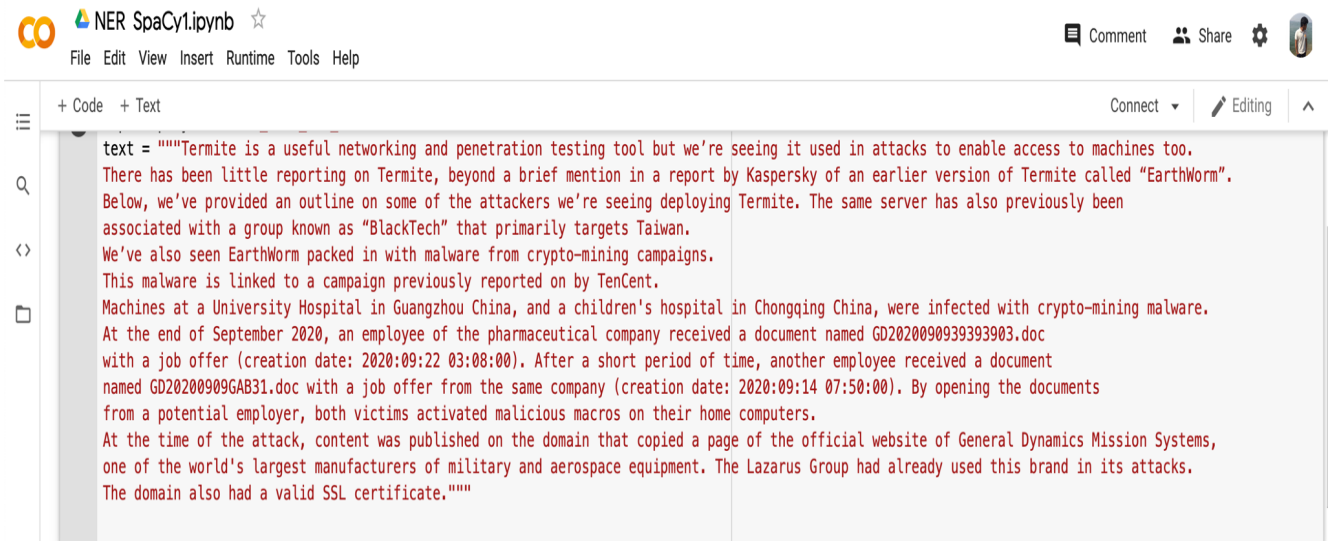


Figure 2: Text compiled from OTX Pulses

Step 2: Performing Named Entity Recognition (NER) on the raw text either using the built in labels or using custom annotations on the raw text:-

i) With built in labels for entities (without custom annotations on text)

I am using **spaCy** with Python on Google Colaboratory to perform Named Entity Recognition on the text acquired in step 1. SpaCy is an open-source software library for advanced natural language processing, written in the programming languages Python.

The "en_core_web_sm" is a trained English pipeline optimized for CPU.

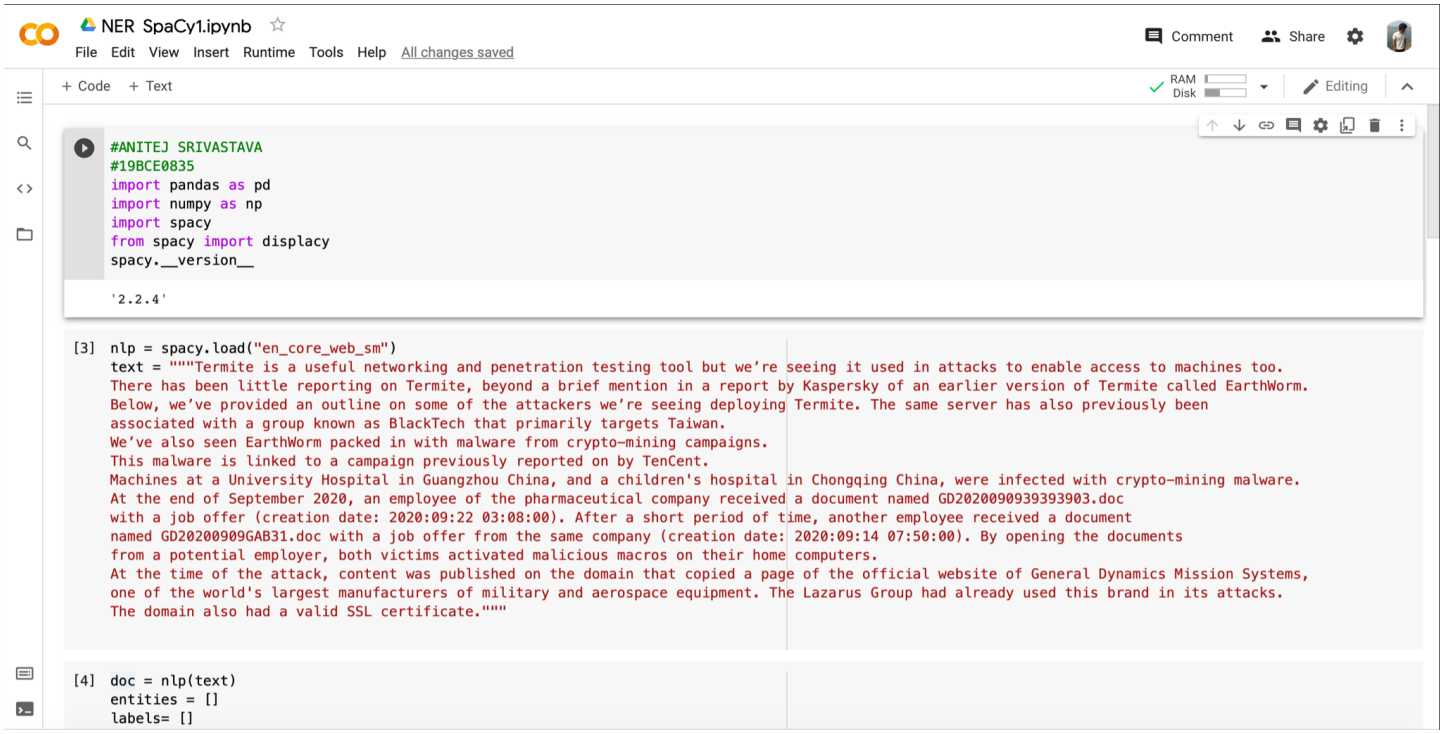
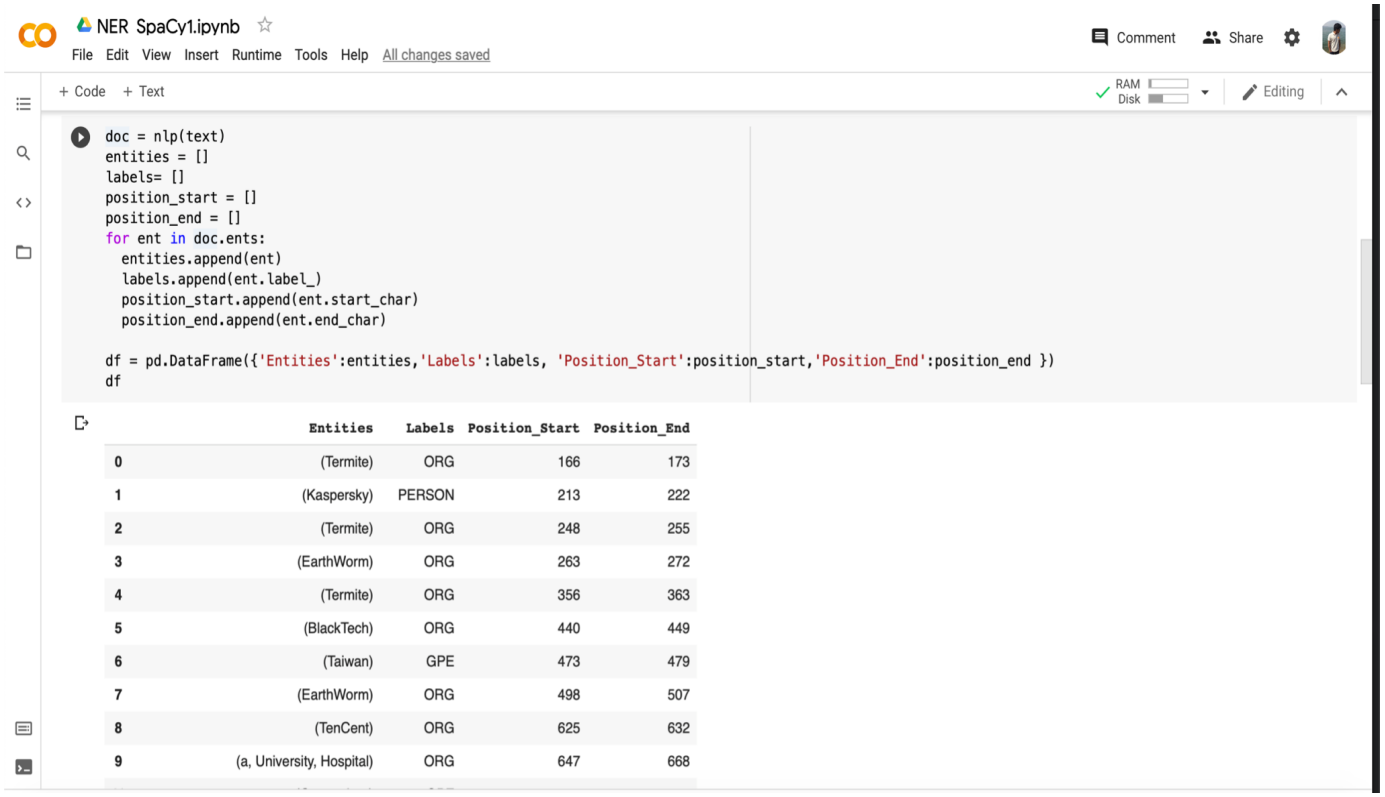
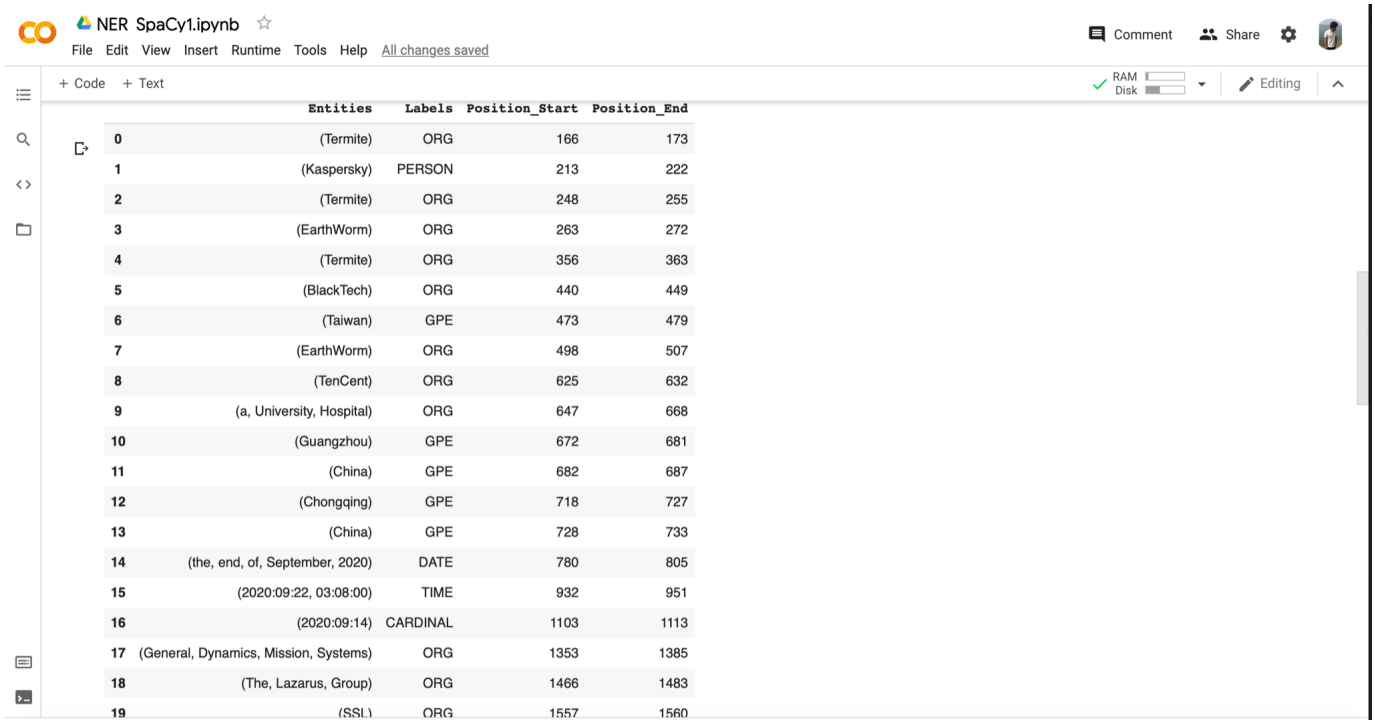


Figure 3: Importing & loading spacy NLP model

The trained NER model then separates the entities it identifies from the raw text into labels from its dictionary.



	Entities	Labels	Position_Start	Position_End
0	(Termite)	ORG	166	173
1	(Kaspersky)	PERSON	213	222
2	(Termite)	ORG	248	255
3	(EarthWorm)	ORG	263	272
4	(Termite)	ORG	356	363
5	(BlackTech)	ORG	440	449
6	(Taiwan)	GPE	473	479
7	(EarthWorm)	ORG	498	507
8	(TenCent)	ORG	625	632
9	(a, University, Hospital)	ORG	647	668



	Entities	Labels	Position_Start	Position_End
0	(Termite)	ORG	166	173
1	(Kaspersky)	PERSON	213	222
2	(Termite)	ORG	248	255
3	(EarthWorm)	ORG	263	272
4	(Termite)	ORG	356	363
5	(BlackTech)	ORG	440	449
6	(Taiwan)	GPE	473	479
7	(EarthWorm)	ORG	498	507
8	(TenCent)	ORG	625	632
9	(a, University, Hospital)	ORG	647	668
10	(Guangzhou)	GPE	672	681
11	(China)	GPE	682	687
12	(Chongqing)	GPE	718	727
13	(China)	GPE	728	733
14	(the, end, of, September, 2020)	DATE	780	805
15	(2020:09:22, 03:08:00)	TIME	932	951
16	(2020:09:14)	CARDINAL	1103	1113
17	(General, Dynamics, Mission, Systems)	ORG	1353	1385
18	(The, Lazarus, Group)	ORG	1466	1483
19	(SSL)	ORG	1557	1560

Figure 4: Generating labels on the raw text

Terminology:

ORG - organisation
GPE - Geopolitical entity
(Rest labels are self-explanatory)

The model is not perfect and has some errors e.g: Kaspersky is incorrectly labelled as PERSON whereas it should have been ORG.

Below is the visualisation of the entire text using **displacy**.

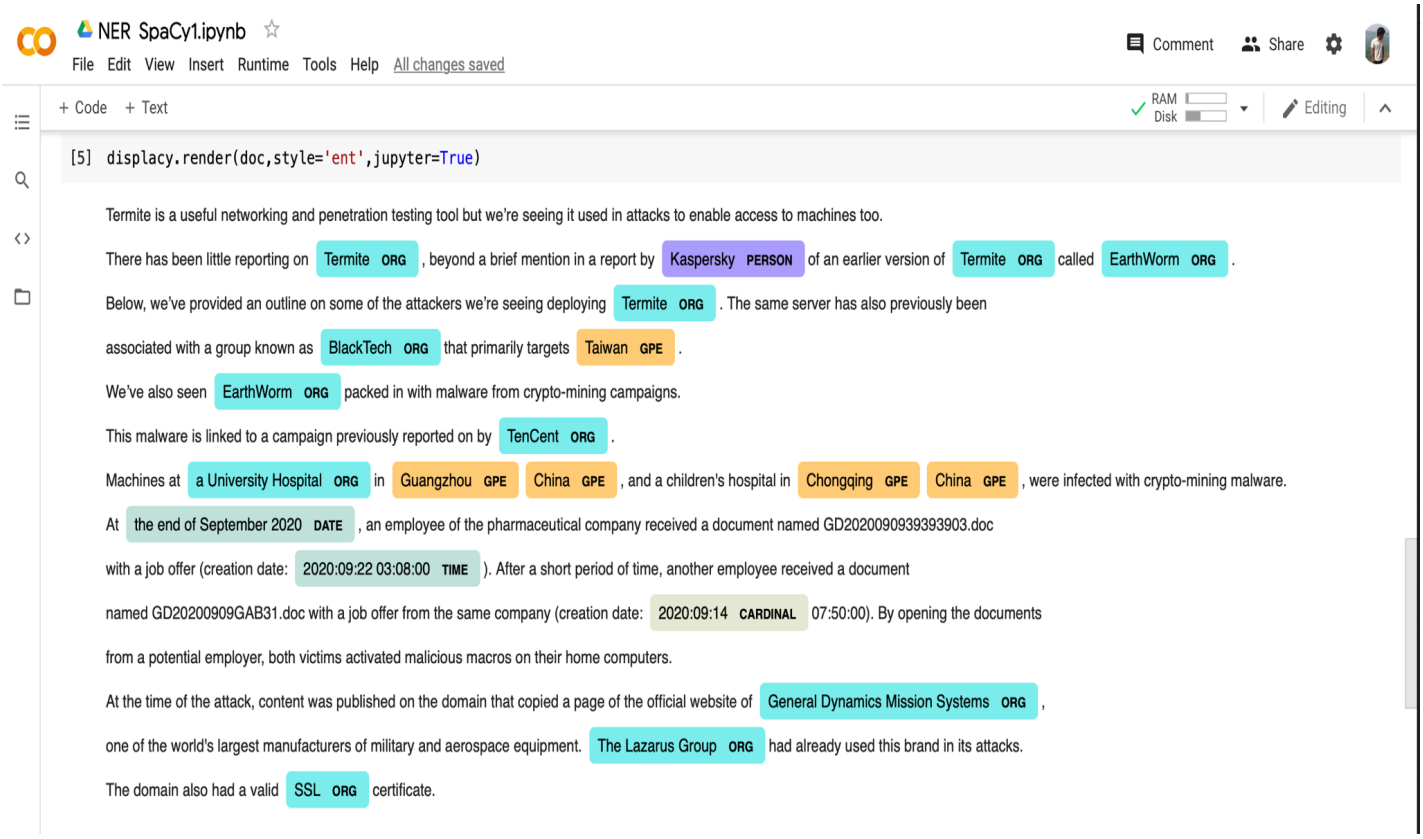


Figure 5: Visualising the results on raw text

ii) With custom labels for the datasets (with custom annotations):

I have defined a data dictionary with custom text annotations for various entitites in the cybersecurity/digital forensics raw text that I extracted from the OTX Pulse.
I intend to train the NER model with the defined data dictionary, so that it can identify custom labels for entitities that are otherwise not built into spaCy.

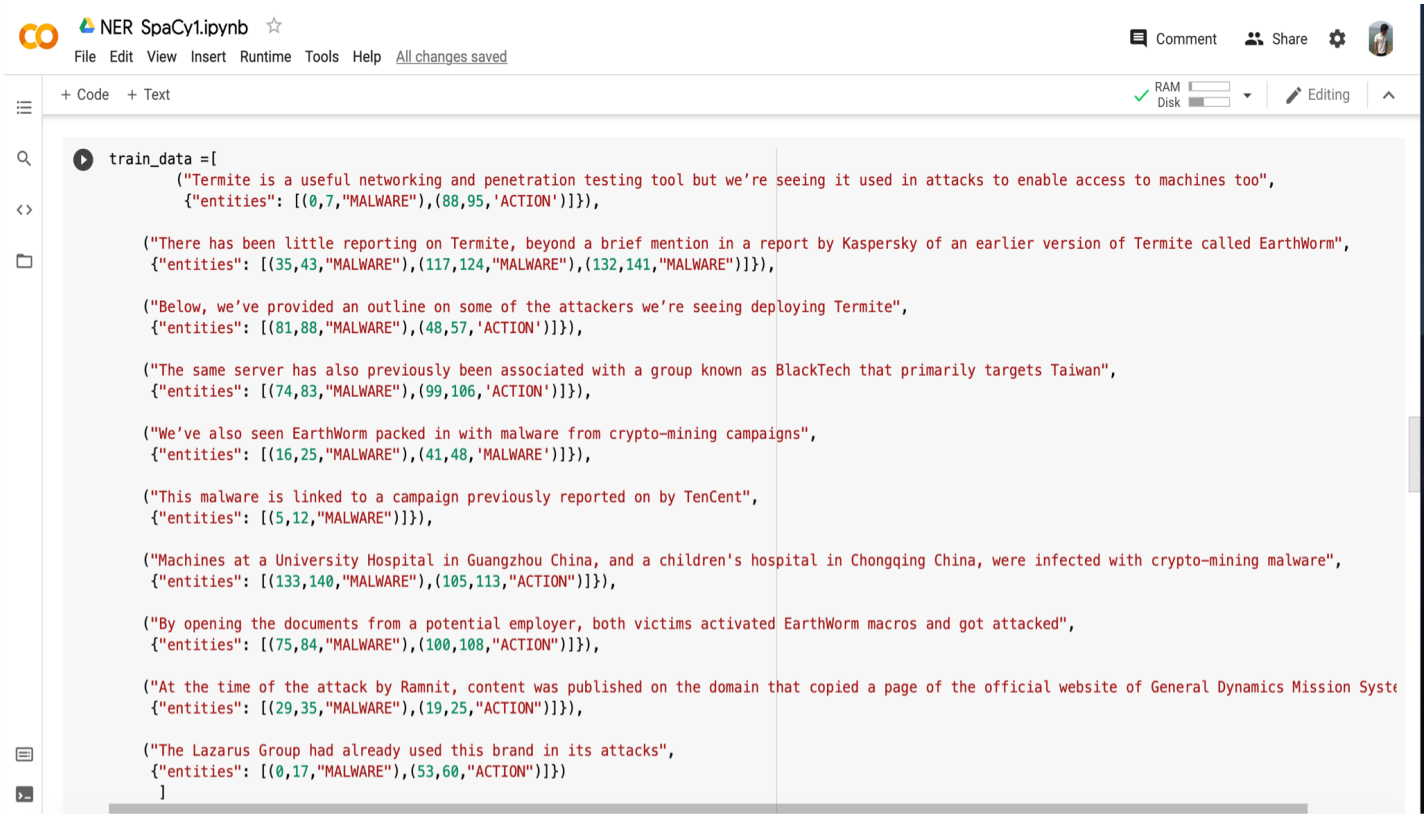


Figure 6: Defining the required dictionary

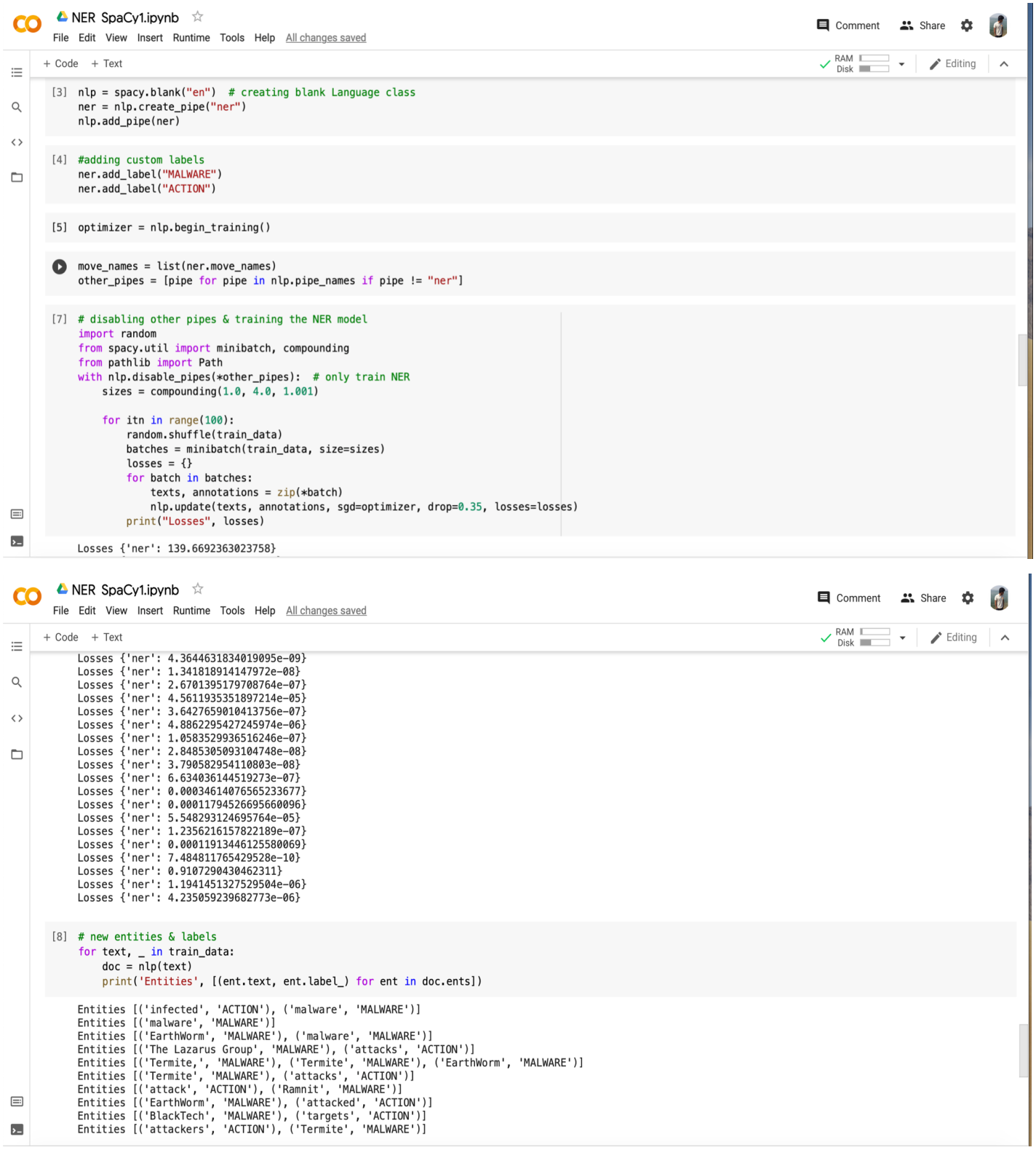


Figure 7: Training the NER model with the dictionary

Below is the output that we get on the testing data,

The results are very close to what I expected:

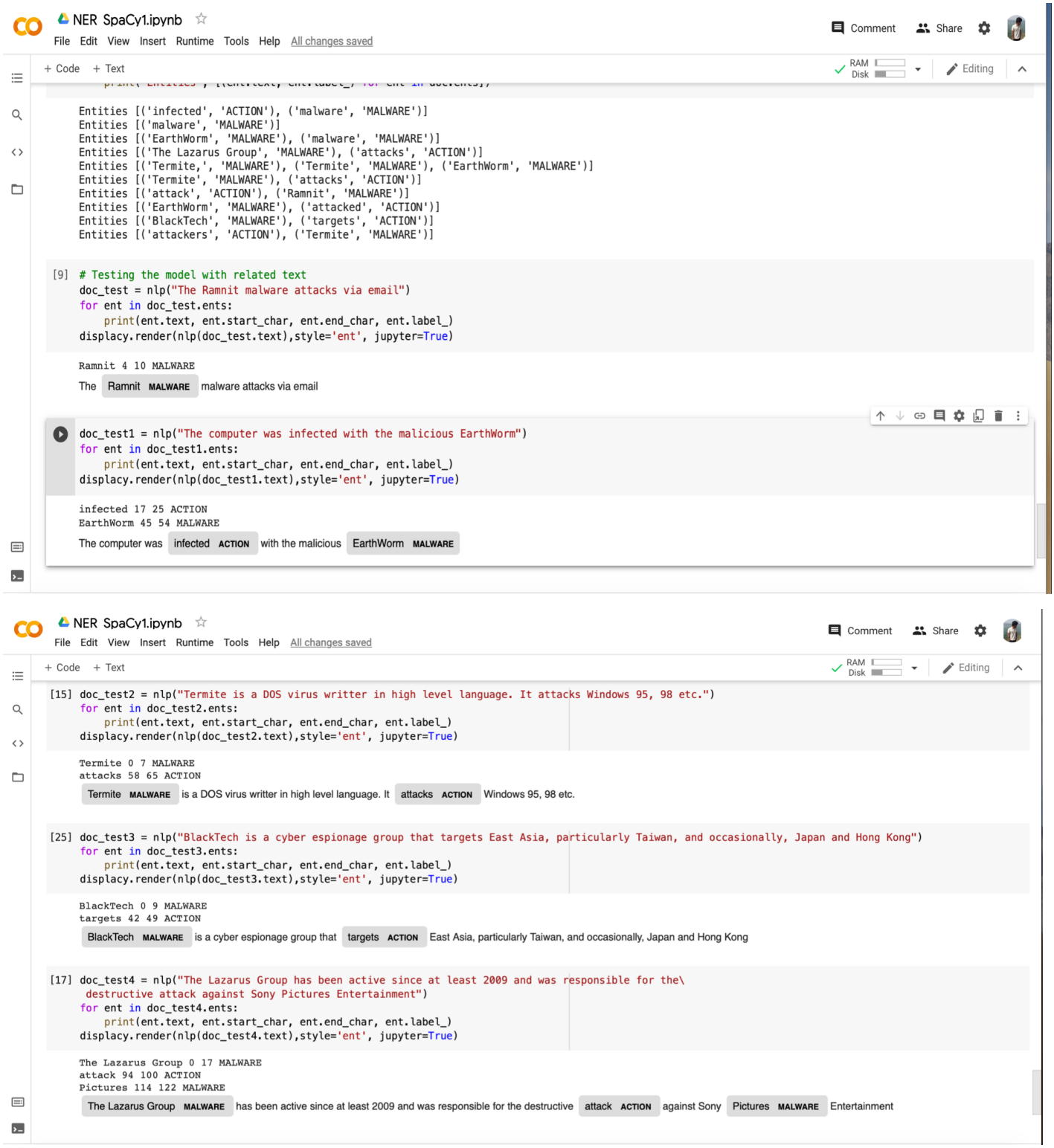
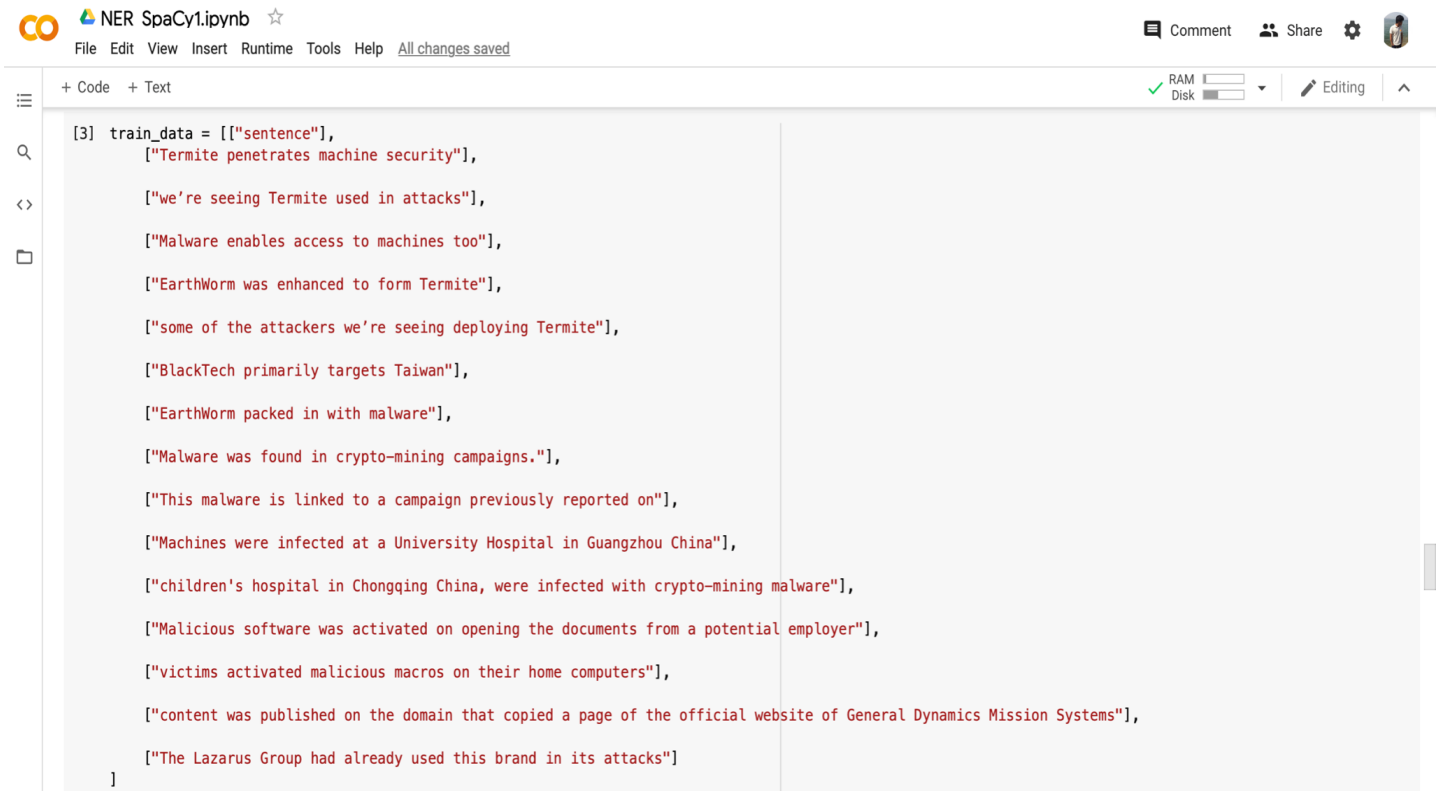


Figure 8: Testing the NER model with random texts

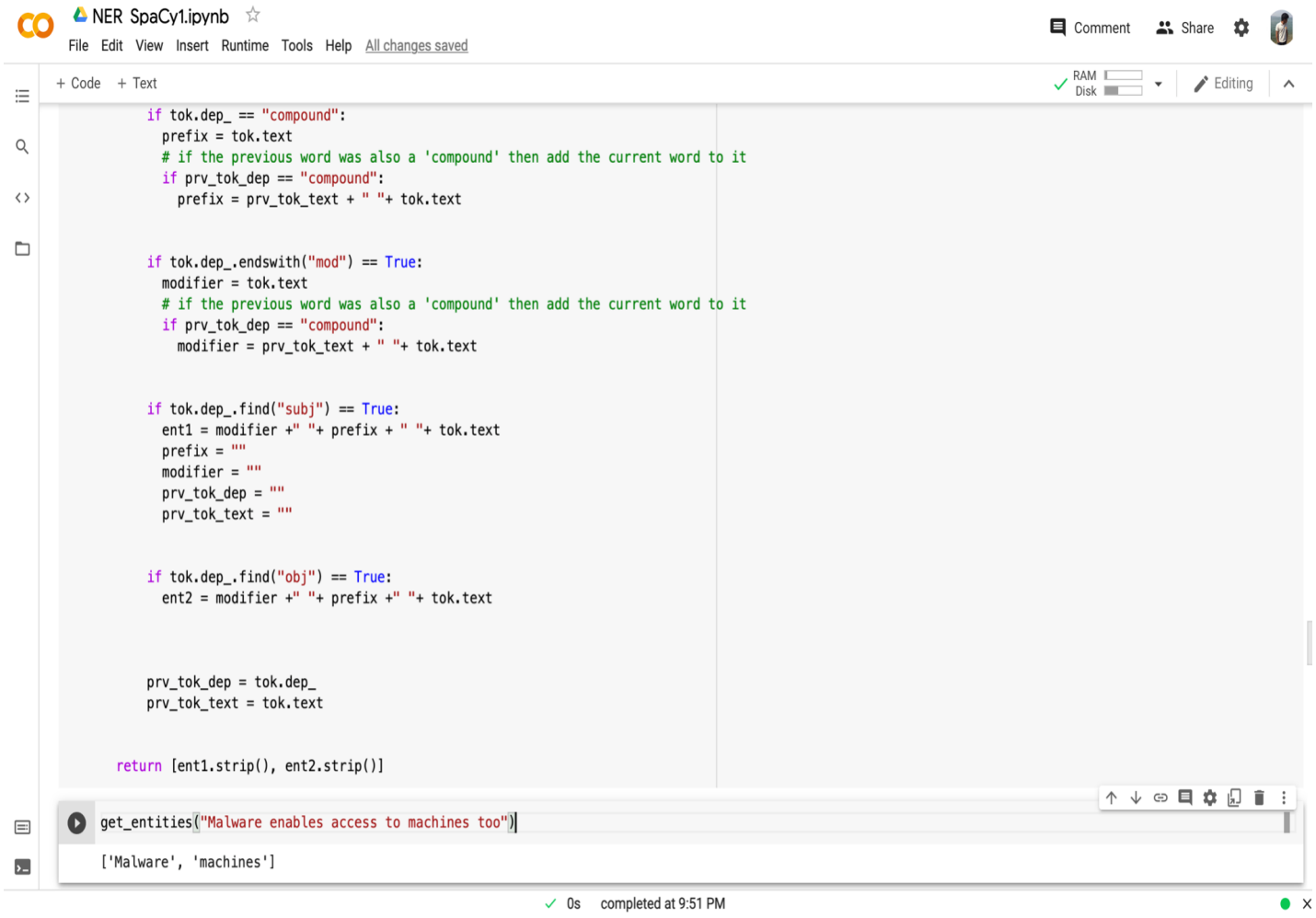
The following is the updated work after uploading the above work on Vtop.

Step 3: As we move towards building a **knowledge graph**, we refine the sentence segmentation of the data corpus collected. The sentences are segmented such that, there is one subject, one object & a ROOT to relate the two.

The subject of the sentence will act as source & object will act as the target with the root as the edge between source & target.

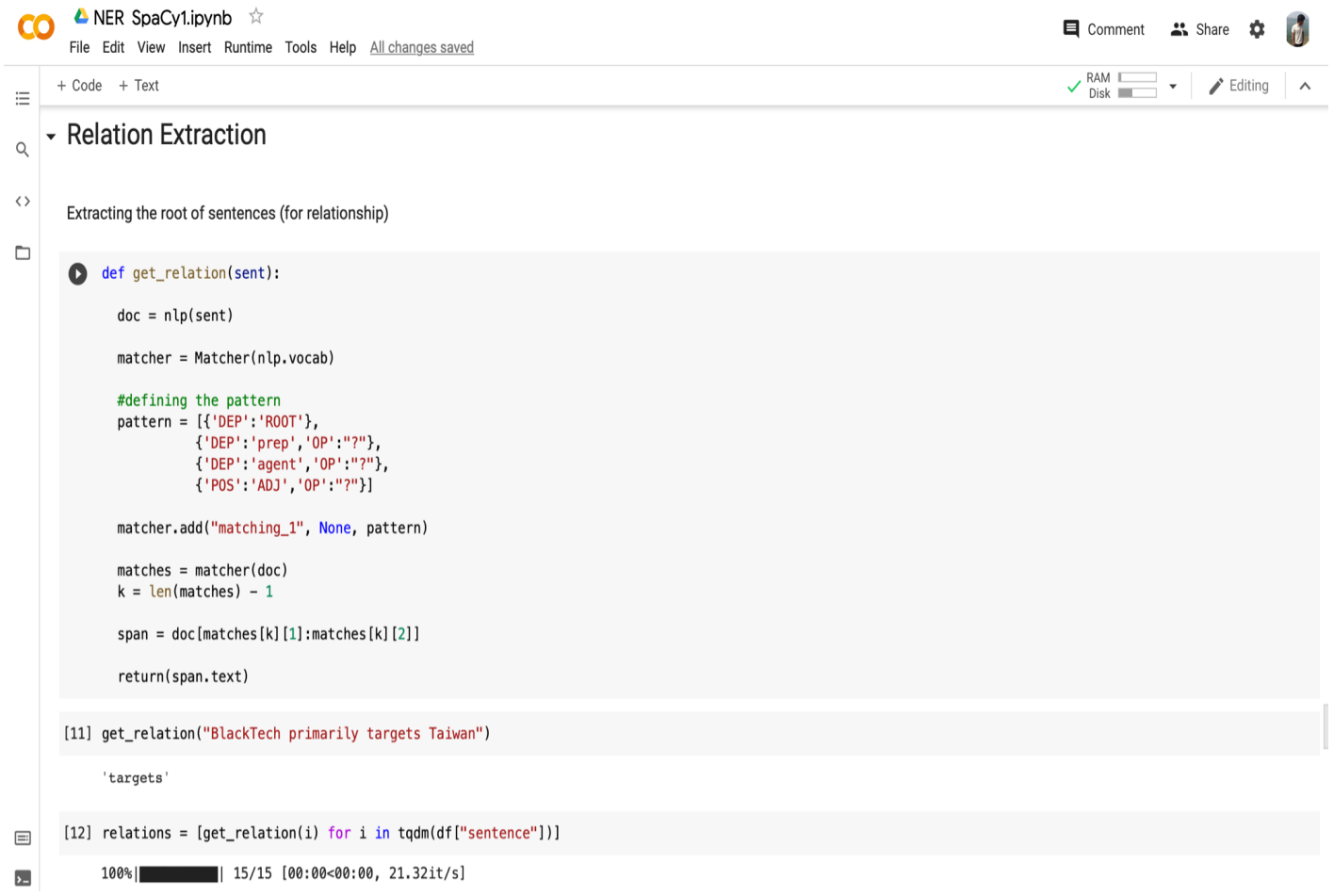


The subject and object entities of the sentences are extracted.



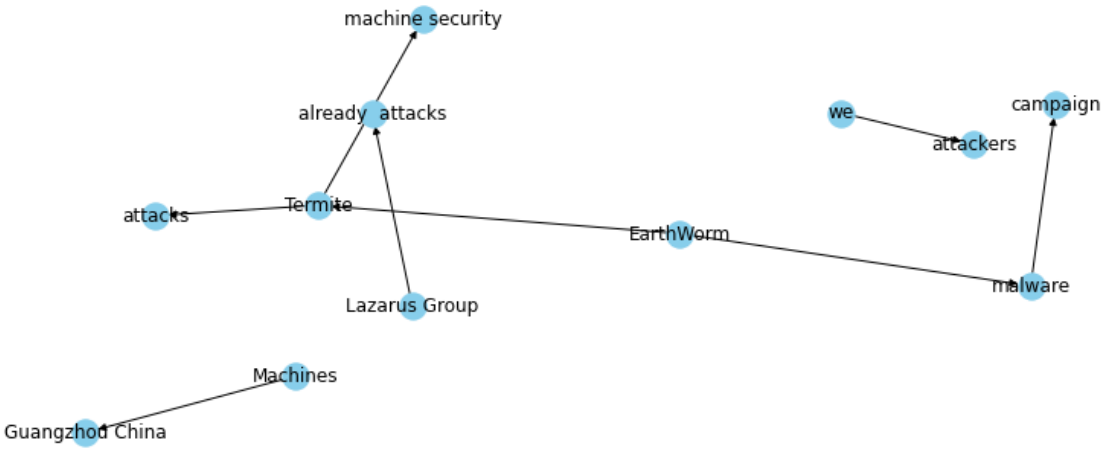
Next the relations are extracted to form the edges of the Graph using spaCy's Matcher, which lets you find words and phrases using rules describing their token attributes.

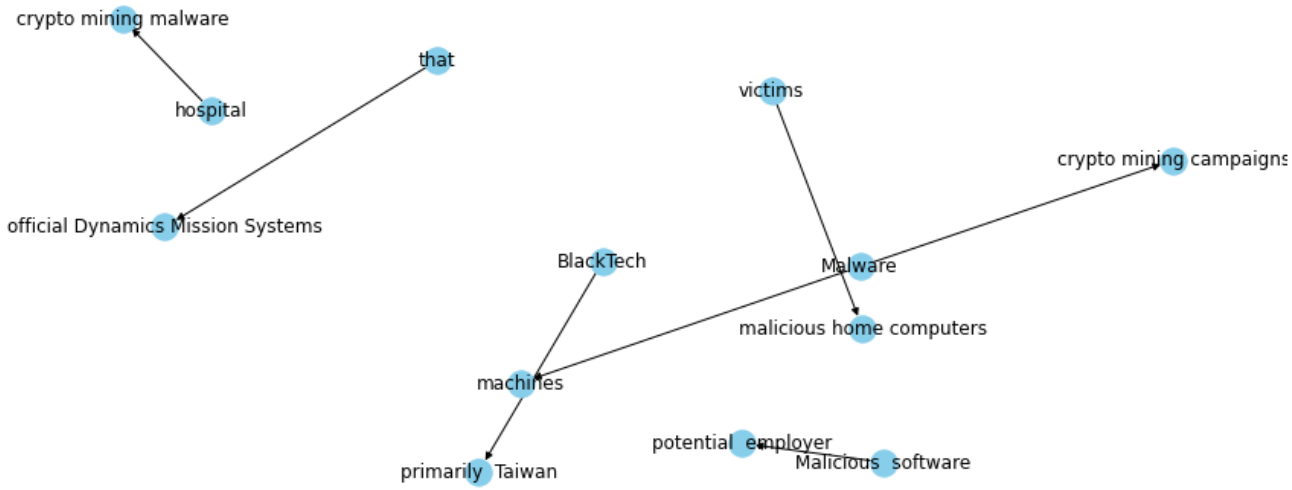
The pattern defined in the function tries to find the ROOT word or the main verb in the sentence.



Finally, using the *networkx* library, created a network from the dataframe in which the source, target nodes & the relation edges are structured & hence the knowledge graph. The **tuned graph** is on the next page.

The nodes represent the entities, and the edges or connections between the nodes represent the relations between the nodes.





Following is the result of tuning the knowledge graph formation:

I have done the tuning in a separate colab file, so as to not alter the previous work:

https://colab.research.google.com/drive/1W6zPpdAHuCcGvK9FsTDhSYFdBqKnQoG2?usp=sharing