Large Language Models
Aplicaciones en Clasificación de Nombres de Dominio

Joaquín Bogado

Agenda

- Quién soy
- Qué estuve haciendo en Chequia
- Qué son los DGAs
- Qué métodos existen para su detección
- Por qué usar LLMs para detectar DGAs (y por qué no)
- Cómo detectar DGAs con LLMs
  - Entrenar un GPT para generar dominios
  - Transformar el generador en un clasificador
- Cómo abordar el estudio de los Large Language Models
- Q & A

Quién soy

Joaquin Bogado

Soy Licenciado en Informática (2014) y Doctor en Ciencias Informáticas (2021) por la Facultad de Informática de UNLP. Tesis de postgrado en Machine Learning y Data Analysis aplicada redes de computadoras (LINTI - IFLP - CERN).

Seleccionado en Jul/2021 para un postdoc en Stratosphere Laboratory (AIC - FEL - ČVUT). En Jul/2022 me mudé a Praga y me volví a principios de Abr/2024.

Durante el postdoc, hice investigaciones relacionadas con ML y DA aplicada a la seguridad informática, en particular sobre el protocolo DNS.

Qué estuve haciendo en Chequia

El Stratosphere Laboratory, dependiente del Artificial Intelligence Center (AIC) en la Facultad de Ingeniería Eléctrica (FEL) en la Universidad Técnica de Chequia en Praga (ČVUT v Praze), centra sus estudios en la intersección entre la inteligencia artificial, la seguridad informática y la ayuda a la comunidad. Tiene proyectos como SLIPS, AIP, CivilSphere, shouldiclick.com y otros tantos.

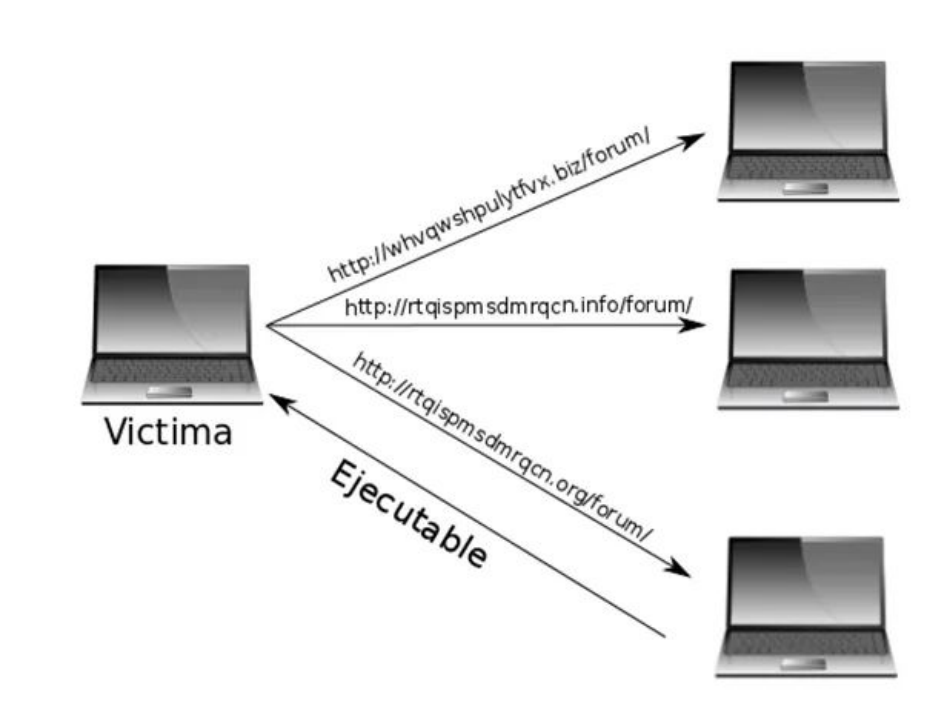
Entre los miembros del equipo cuenta a Sebastián García, Verónica Valeros, Maria Rigaki y Ondra Lucas.

Uno de los proyectos en los que participé y que viene al caso para esta charla fue DNS4EU.

Qué son los DGAs

- Los Domain Generation Algorithms son algoritmos para generar nombres de dominio de manera pseudoaleatoria.
- Son parte del mecanismo de comunicación entre el malware y las víctimas que sirve para evitar hardcodear el nombre de dominio en el fuente del virus.
- Los nombres de dominio se generan aleatoriamente, pero en orden. El malware trata de establecer comunicación con cada uno de los dominios de a uno a la vez. El atacante solo debe registrar un dominio de la lista.
- Es muy difícil de bloquear sin saber la lista de dominios que genera el malware.
- Hay varias familias, algunas más difíciles de detectar que otras.

Qué son los DGAs



alureon
bamital
banjori
bazar
bedep
beebone
blackhole
bobax
ccleaner
charbot
chinad
chir
conficker
corebot
cryptolocker
darkshell
deception
diamondfox
dircrypt
dmsniff
dnsbenchmark
dnschanger
downloader
dyre
ebury
ekforward
emotet
feodo
fobber
fosniw
gameover
gozi
goznym
gspy
hesperbot
infy
kraken
legit
locky
m0yv
madmax
makloader
matsnu
mirai
modpack
monerodownloader
monerominer
murofet
murofetweekly
mydoom
necurs
newgoz
nymaim
nymaim2
oderoor
omexo
orchard
padcrypt
pandabanker
pitou
pizd
proslikefan
pushdo
pushdotid
pykspa
pykspa_noise
qadars
qakbot
qhost
qsnatch
ramdo
ramnit
ranbyus
randomloader
reconyc
redyms
rovnix
sharkbot
shifu
shiotob
simda
sisron
sphinx
suppobox
sutra
symmi
szribi
tempedreve
tempedrevetdd
tinba
tinynuke
tofsee
torpig
tsifiri
tufik
ud2
ud3
ud4
unnamed_downloader
urlzone
vawtrak
verblecon
vidro
vidrotid
virut
volatilecedar
wd
xshellghost
xxhex
zeus-newgoz
zloader

Qué métodos existen para su detección

- Network Behaviour
  - Número anómalo de mensajes NXDOMAIN
  - Conexiones a dominios poco frecuentes
  - Mensajes TXT
- String Analysis
  - Expresiones regulares
  - DGA decompiling
  - Análisis estadístico (entropía, frecuencia, ratios, etc)
  - Procesamiento del lenguaje natural (part-of-speech tagging, token analysis, etc)
  - Análisis semántico (LLMs 🤞)

Por qué usar LLMs para detectar DGAs (y por qué no)

Pros:

- Más efectivos en familias difíciles basadas en texto (charbot, suppobox, matsnu)
- Son genéricos (no específicos para una familia)
- Gran poder de comprensión (compresión?) del significado del texto

Cons:

- Los modelos comerciales son inescrutables (y los open source?)
- Los modelos comerciales pueden ser costosos
- Pueden ser lentos para algunos casos de uso
- Puede ser difícil implementar una solución

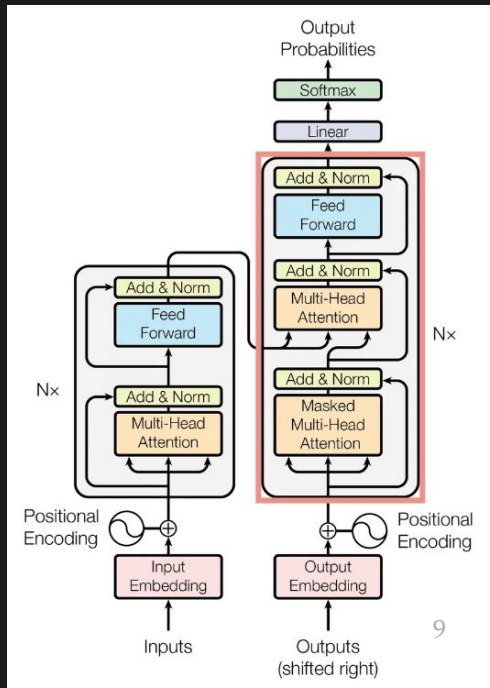
Cómo detectar DGAs con LLMs - parte 1

- ➢ Implementar un Generative Pre-trained Transformer (decoder only)
- ➢ Entrenarlo usando nombres de dominios legítimos (TRANCO list)
- ➢ Enjoy!

```
In [1]: import tiktoken

In [2]: tokenizer = tiktoken.get_encoding('gpt2')

In [3]: tokenizer.encode('lifia.unlp.edu.ar')
Out[3]: [36195, 544, 13, 403, 34431, 13, 15532, 13, 283]

In [4]: [tokenizer.decode([x]) for x in tokenizer.encode('lifia.unlp.edu.ar')]
Out[4]: ['lif', 'ia', '.', 'un', 'lp', '.', 'edu', '.', 'ar']
```

☝

Byte-Pair Encoding



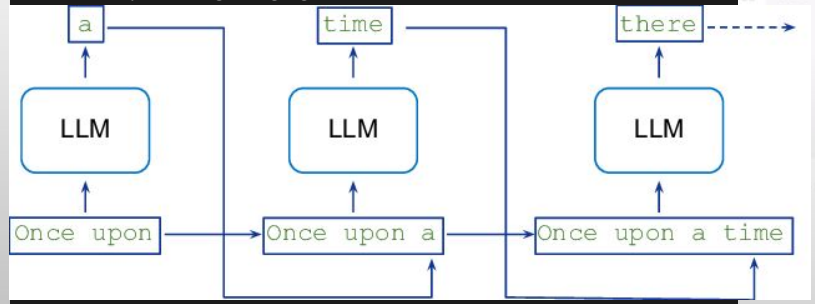
Cómo detectar DGAs con LLMs - parte 1

- ¿Cómo procesan los LLMs texto de longitud variable?
- ¿Cómo les parece que son los ejemplos de entrenamiento?

https://youtu.be/kCc8FmEb1nY?feature=shared&t=1027

```
emma
... ---> e
..e ---> m
.em ---> m
emm ---> a
mma ---> .
olivia
... ---> o
..o ---> l
.ol ---> i
oli ---> v
liv ---> i
ivi ---> a
via ---> .
ava
... ---> a
..a ---> v
.av ---> a
ava ---> .
isabella
... ---> i
..i ---> s
```

Shanahan, Murray & McDonell, Kyle & Reynolds, Laria. (2023). Role-Play with Large Language Models.



https://youtu.be/TCH_1BHY58I?feature=shared&t=609

```
block_size = 8
train_data[:block_size+1]
```

```
tensor([18, 47, 56, 57, 58,  1, 15, 47, 58])
```

```
x = train_data[:block_size]
y = train_data[1:block_size+1]
for t in range(block_size):
    context = x[:t+1]
    target = y[t]
    print(f"when input is {context} the target: {target}")
```

```
when input is tensor([18]) the target: 47
when input is tensor([18, 47]) the target: 56
when input is tensor([18, 47, 56]) the target: 57
when input is tensor([18, 47, 56, 57]) the target: 58
when input is tensor([18, 47, 56, 57, 58]) the target: 1
when input is tensor([18, 47, 56, 57, 58,  1]) the target: 15
when input is tensor([18, 47, 56, 57, 58,  1, 15]) the target: 47
when input is tensor([18, 47, 56, 57, 58,  1, 15, 47]) the target: 58
```

Cómo detectar DGAs con LLMs - parte 1

Una vez que uno entrena un GPT para generar dominios, lo que se obtiene en la práctica es un DGA. Entrenado para generar dominios similares a los legítimos, de a un token a la vez.

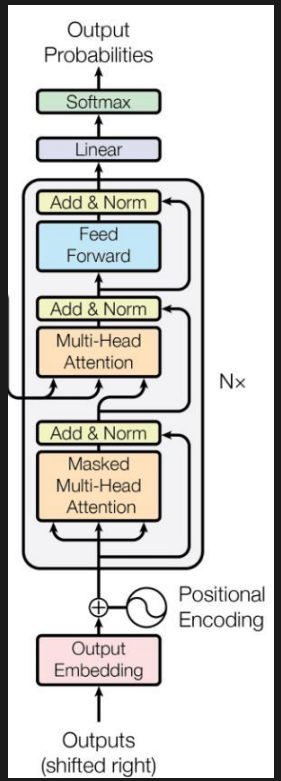
Acá me llevé algunas sorpresas.

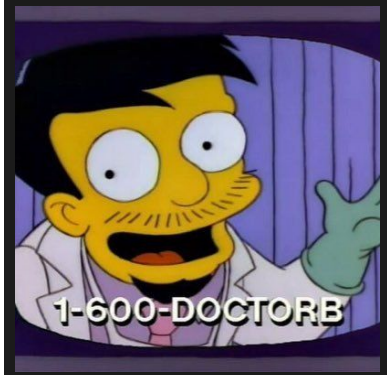
Incluso un modelo pequeño (6M) aprende a generar dominios punycode válidos.

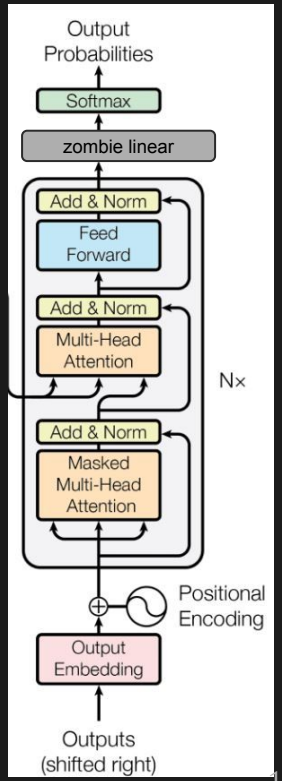
> xn--4dbcagd2c0b2bce3h.xn--wgbl6a → संगठन.भारत
>
> xn--d1abbgf6aiiy.xn--p1ai → pф.pyc

Cómo detectar DGAs con LLMs - parte 2

- Dos formas de hacer fine-tuning
  - Reentrenar todos los pesos del modelo
  - Reemplazar algunas capas del modelo (generalmente, las últimas)
- Yo opté por emplazar la última capa del modelo.
- Entrenar la capa usando los embeddings del modelo para predecir DGA/Legit (binary) o las familias de DGAs (multi-class).
- De nuevo acá hay varias posibles soluciones dependiendo de las necesidades de la tarea de clasificación.







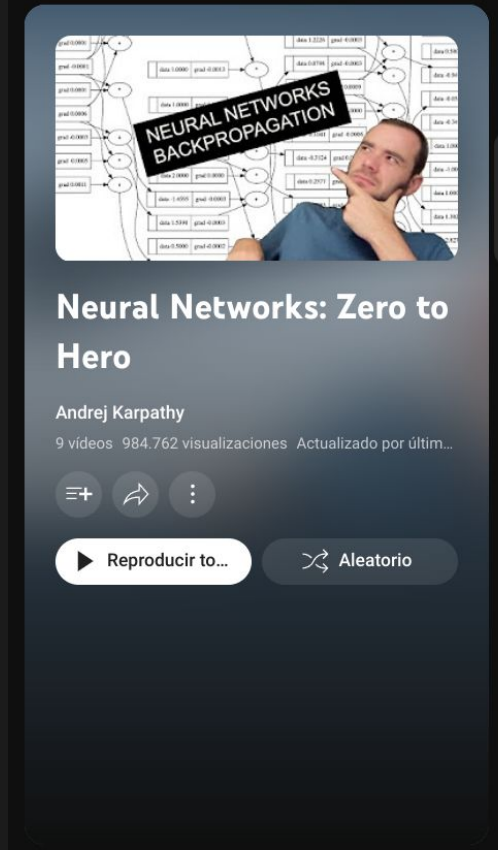
Cómo abordar el estudio de los Large Language Models

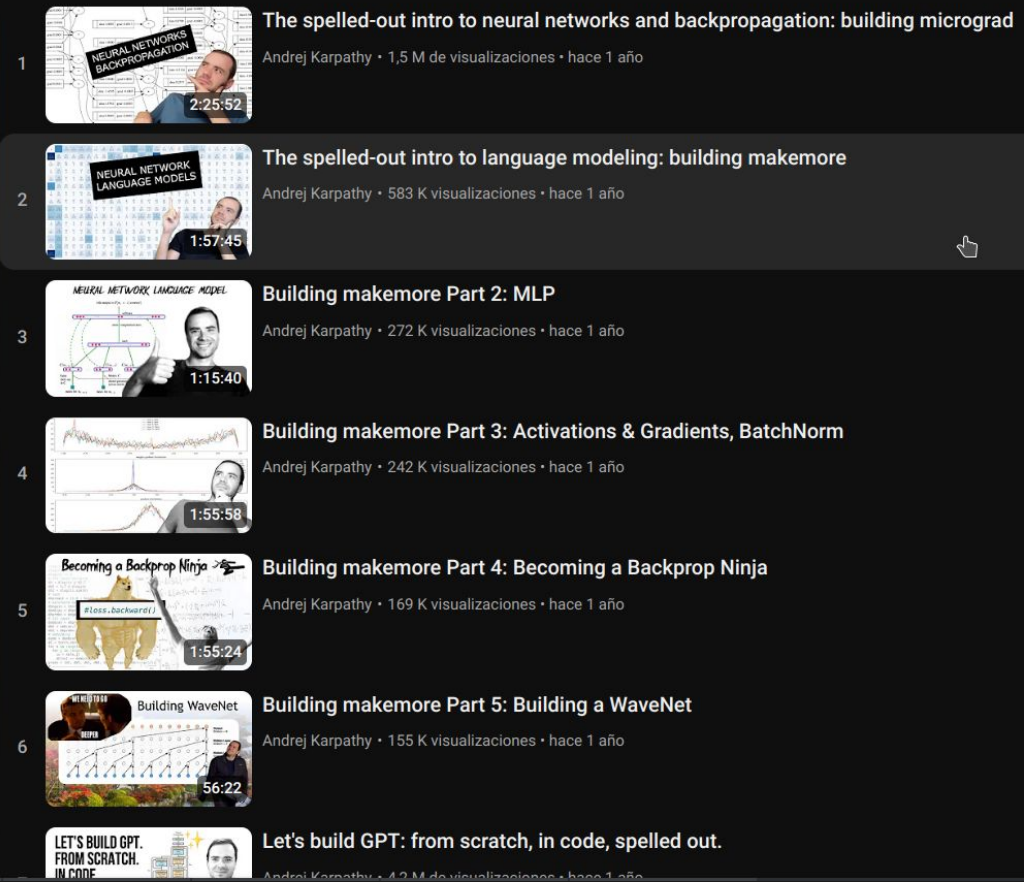
- Conocer los componentes básicos de la arquitectura y la nomenclatura.
  - Tokens, Embeddings (identity, positional), Normalization (batch vs layer), Attention mechanisms (self, cross Att), Transformers blocks, Encoders vs Decoders, Loss functions (NLogL y Cross Entropy), Logits vs probabilities. ⇐ *La lista no es exhaustiva, perdón.*
- No alcanza conocer cuales son, hay que conocer su función. ¿Por qué es necesaria la capa de LayerNorm? ¿Por qué hacen falta dos embeddings? ¿Cuáles son los efectos de la cuantización?
- Objetivos a nivel researcher
  - Comprender los beneficios de una técnica nueva
  - Poder diferenciar entre arquitecturas
  - Identificar problemas en el state-of-the-art
- Objetivos a nivel user
  - Comprender las capacidades y limitaciones de los modelos

Q & A

Backup

Recursos





https://www.youtube.com/playlist?list=PLAqhIrjkxbuWI23v9cThsA9GvCAUhRvKZ

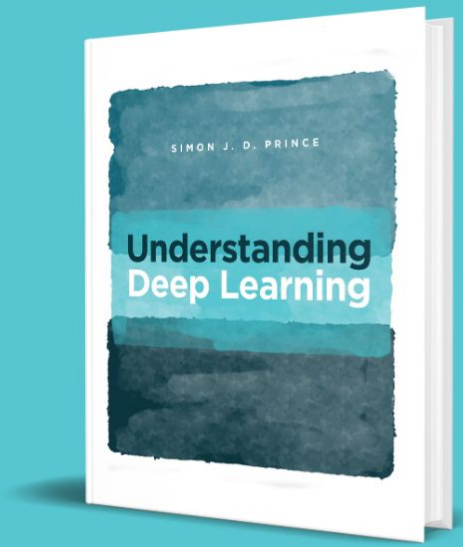
Recursos

Understanding Deep Learning

Notebooks Instructors Media More

RECENT NEWS:

03/12/24	Book now available again.
02/21/24	New blog about the Neural Tangent Kernel.
02/15/24	First printing of book has sold out in most places. Second printing available mid-March.
01/29/24	New blog about gradient flow published.
12/26/23	Machine Learning Street Talk podcast discussing book.
12/19/23	Deeper Insights podcast discussing book.
12/06/23	I did an interview discussing the book with Borealis AI.
12/05/23	Book released by The MIT Press.



CITATION:

```
@book{prince2023understanding,
  author = "Simon J.D. Prince",
  title = "Understanding Deep Learning",
  publisher = "The MIT Press",
  year = 2023,
  url = "http://udlbook.com"}
```

Download full pdf (01 May 2024)

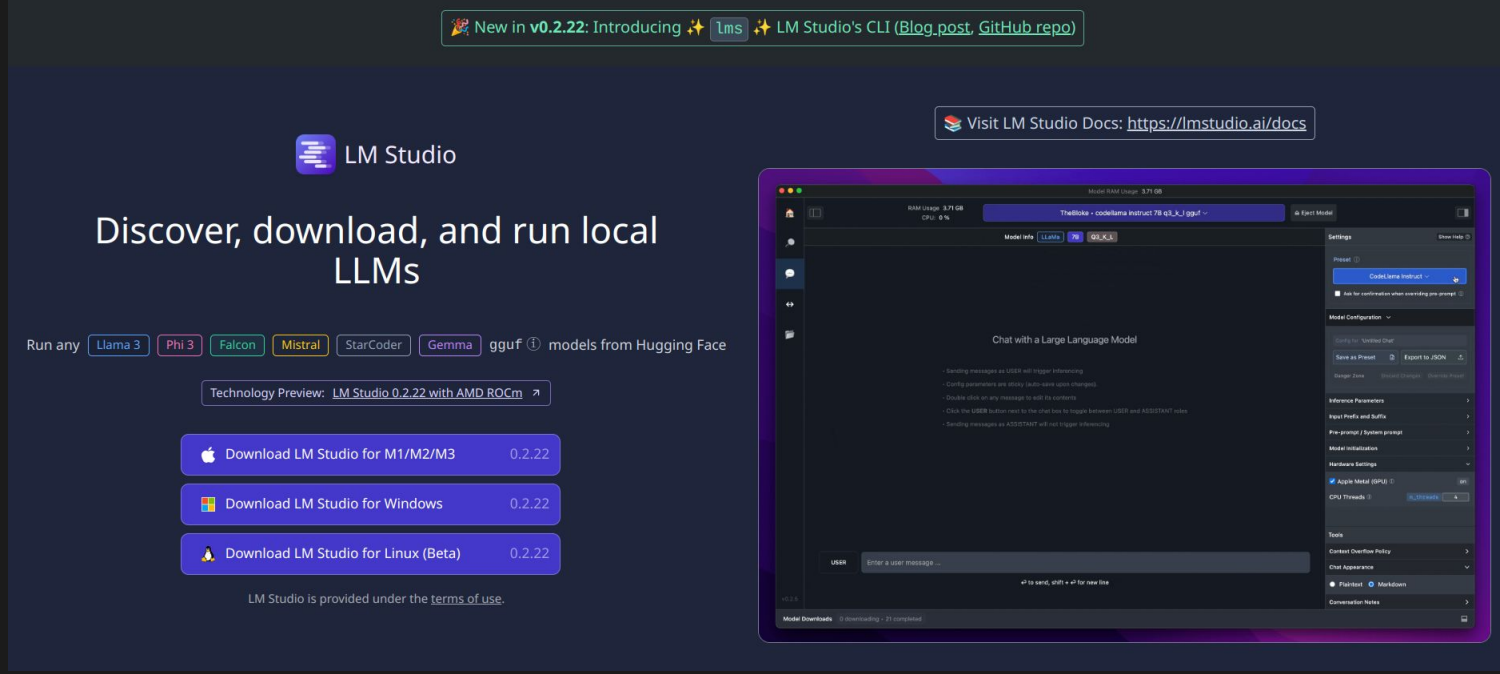
downloads 285k

Buy the book

Answers to selected questions

https://udlbook.github.io/udlbook/

Recursos



https://lmstudio.ai/