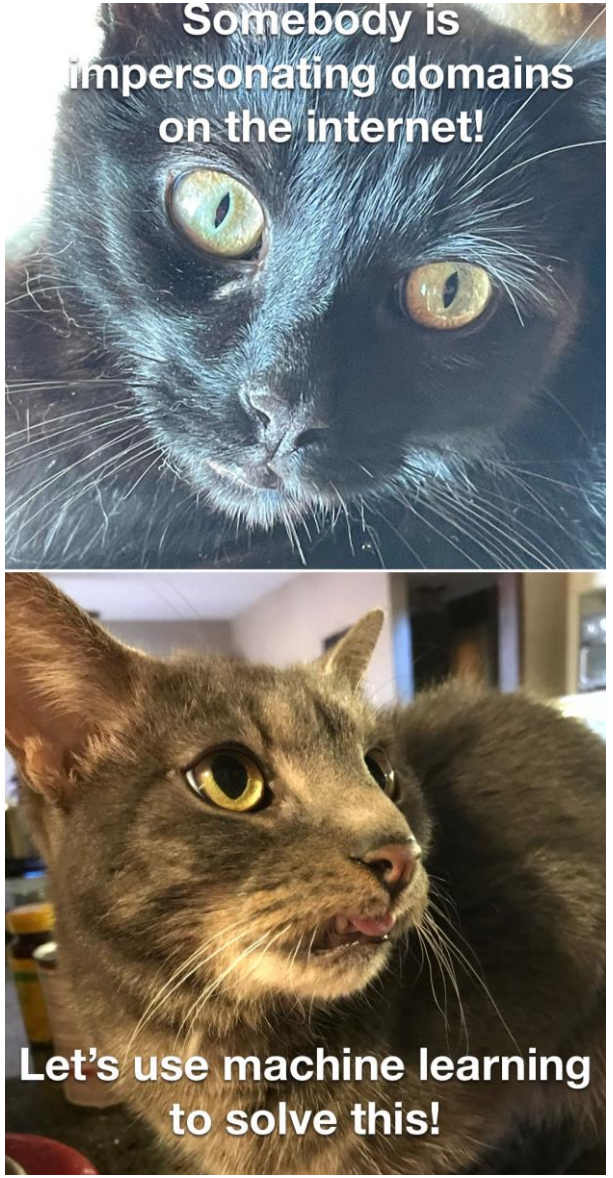
Microsoft Security

# Detecting Homoglyph Domains with LSTMs

Rob Brandon

# What is a homoglyph?

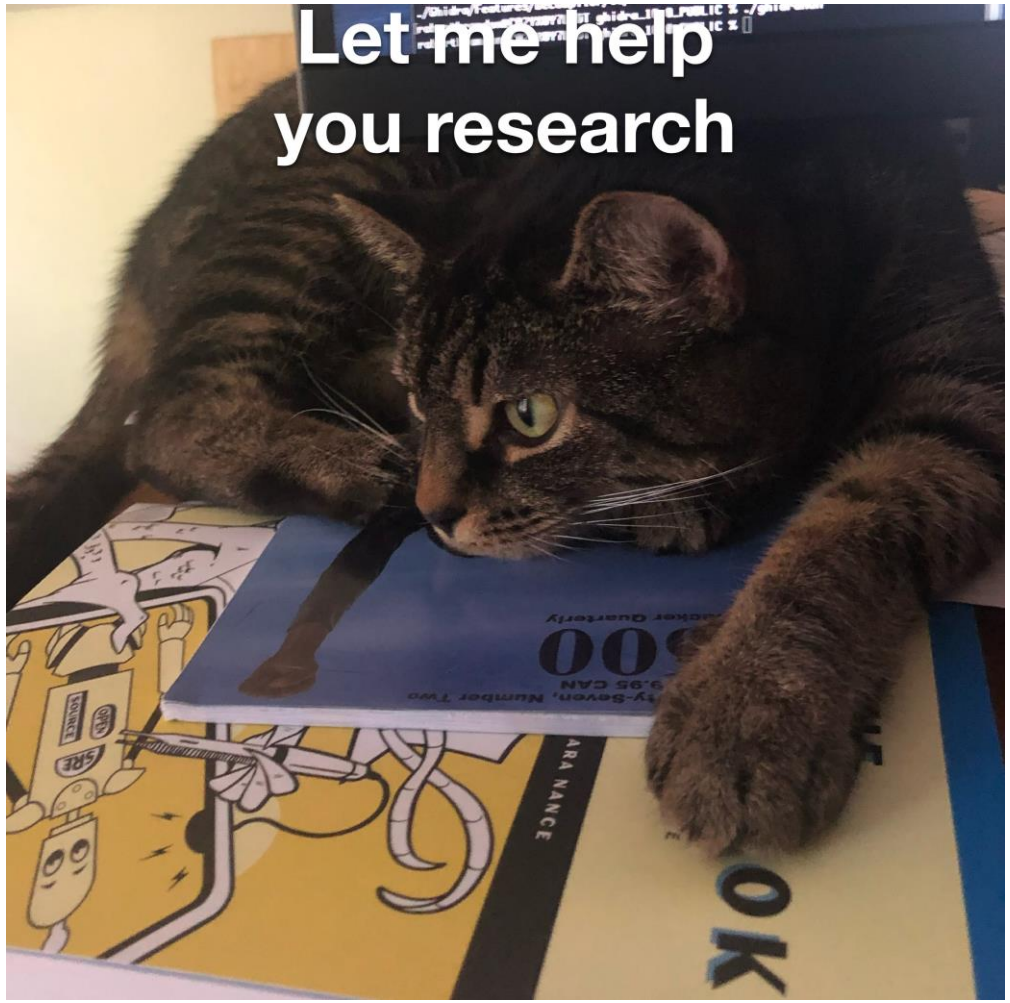
- Homoglyph attacks attempt to fool the human visual system by using glyphs that appear similar to glyphs in the target.
- Can be simple character swaps/substitutions or can be more complicated
  - microsoft.com vs rnicrosoft.com
  - linkedin123.com vs linkedin.com
  - bingproservices vs bing.com
- Used in a variety of attacks by a wide range of attackers for a number of different purposes, i.e. phishing, malware C&C, business email compromise…



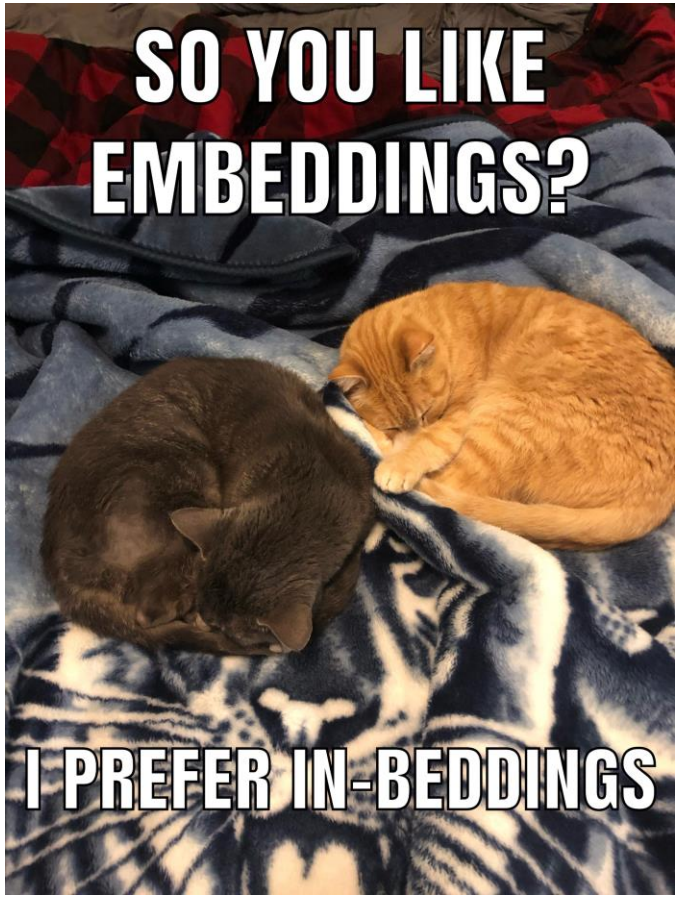
# Historical Approaches

- Lots of distance metrics:
  - Levenstein, n-gram comparisons, etc
  - NxN comparisons are not tractable beyond trivial data sets

- Pioneering work using Siamese CNNs by Woodbridge et al
  - Works ok, appears to largely be learning to group things by length and number of dark pixels rather than learning language constructs
  - Task as posed to the network is harder than required



# Embeddings are usually good, lets do that!




- Instead of a CNN over an image of the domain, generate an embedding using sequences of character images
- Propensity of neural networks to memorize training data is actually an advantage here

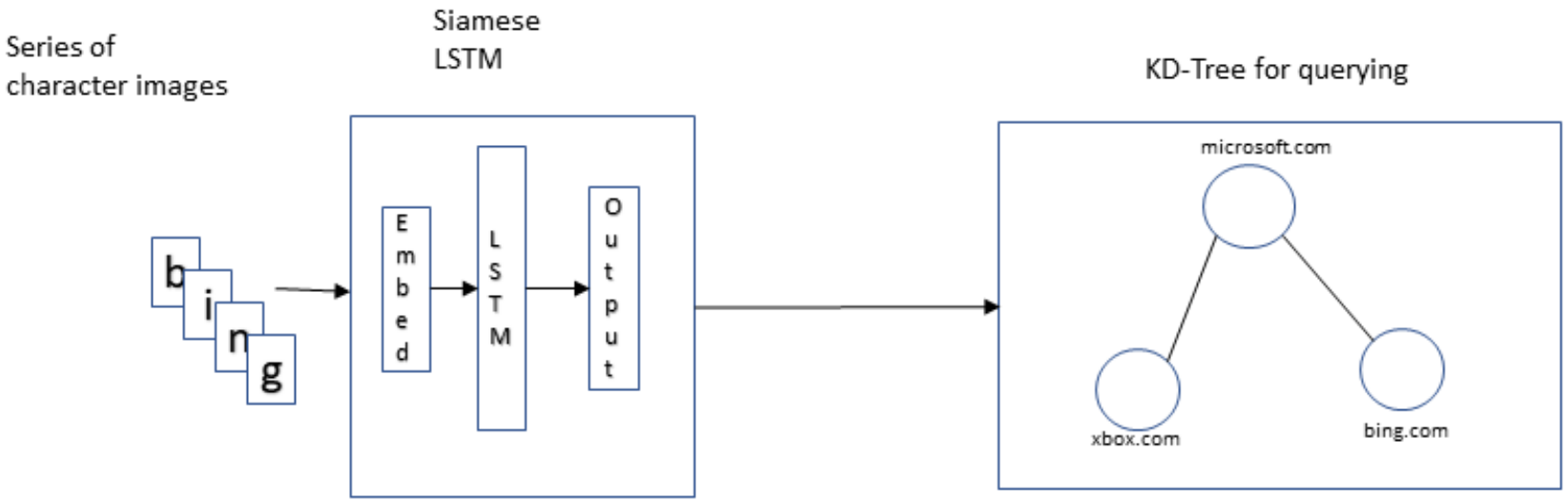
# Data sets

- Like most other problems, data sets are non-existent
- Unlike most other problems, we can easily create our own!

- Lots of data sets for legit domains – in this case we used the Majestic Million dataset available from www.majestic.com

- Worked with several SMEs to build a custom homoglyph generator using homoglyph creation techniques observed in the wild

# Obligatory network diagram



# How to measure effectiveness?

- Common problem for generative and self-supervised models
- NN learns via contrastive loss, but contrastive loss does not actually measure how effective the NN is at embedding domains for the actual problem

- How to measure embedding effectiveness generally is still very much an open problem

# Homoglyphs are a hard problem for people

- Given the purely hypothetical domain www.tailspintoys.com, which of the following hypothetical legitimate domains could it be a homoglyph for?
  - www.tallspintoys.com
  - www.tailspinstoys.com
  - www.talispintoys.com
  - www.tailspintoysllc.com

# The answer is: All of the above

# Accuracy – exact and fuzzy

- Exact match – Desired domain is the closest neighbor in the embedding space to the desired domain
- Fuzzy match – Desired domain is within *n* nearest neighbors, for some value of *n*

# So how good is this approach?

- Model trained on Majestic Million with n=30 for fuzzy accuracy
- Testing set consisted of a single perturbation for each domain
  - Perfect accuracy – 11%
  - Fuzzy accuracy – 22%


- Note: n=30 is arbitrary, fuzzy accuracy increases as n increases

# Does learning transfer?

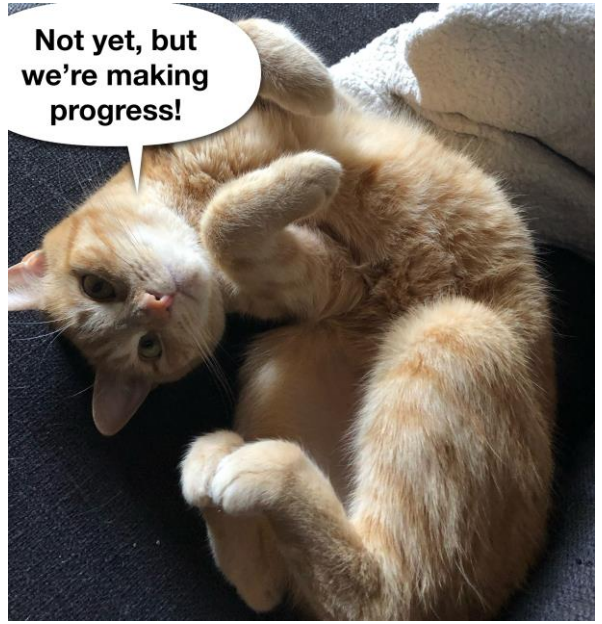
- Basic structure of domains does not change from one environment to the next, so should be able to use model trained on open source domains for generating embeddings of new domains

# Quick transfer test

- Tested against a set 4000 of internal domains that were not in the training set
- No fine-tuning was done was done to the model
  - Perfect accuracy – 15%
  - Fuzzy accuracy – 30%
- Results are even better than the training set!
- Likely that search space size significantly affects accuracy, needs more testing to see how much

# Final thoughts


- Still plenty of work to do fine-tuning the approach
- Current results are still good enough for initial operational work
- Probably works even better if you have a smaller set of domains to monitor



# Questions?