Dns2Vec: Exploring Internet Domain Names Through Deep Learning

Amit Arora

Advisory Engineer, Data Scientist

Hughes Network Systems

HUGHES

An EchoStar Company

ScAINet '19

Resources

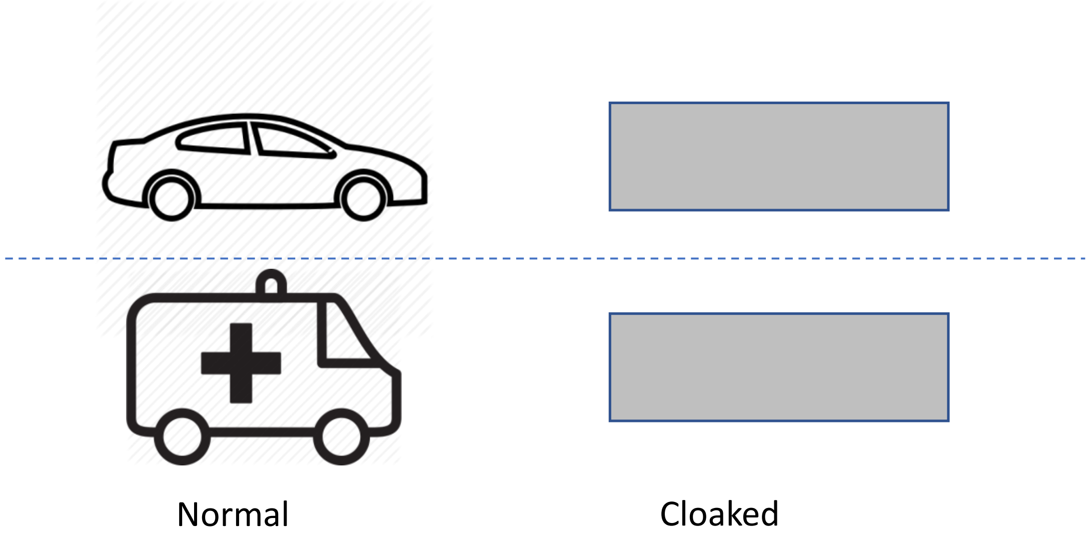
https://github.com/aarora79/dns2vec	Complete source code and paper
https://arxiv.org/pdf/1310.4546.pdf	The original Word2Vec paper
https://github.com/facebookresearch/StarSpace	Facebook StarSpace: Embed all the things!
https://github.com/deezer/w2v_reco_hyperparameters_matter	Word2Vec applied to recommendations, hyperparameters matter
https://radimrehurek.com/gensim/models/word2vec.html	Gensim Word2Vec example
aarora79 aarora79	About me

Problem Statement

Internet Traffic Classification for Large ISPs is a hard problem to solve!

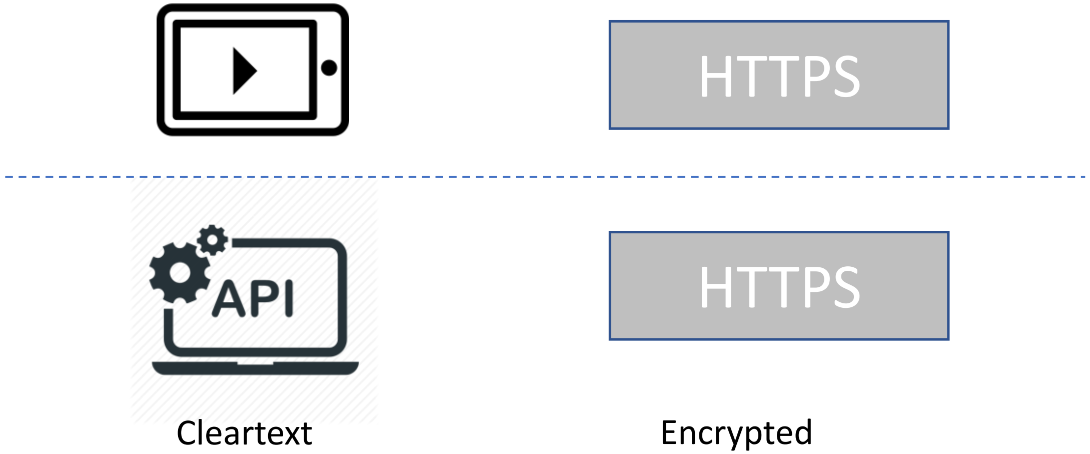
Problem Statement

- Imagine vehicles on a highway, we want to prioritize their movement BUT..
  - They are all wearing cloak, so they all look a like
  - We don’t know how long they are going to be on the road
  - All we know is their starting location and destination



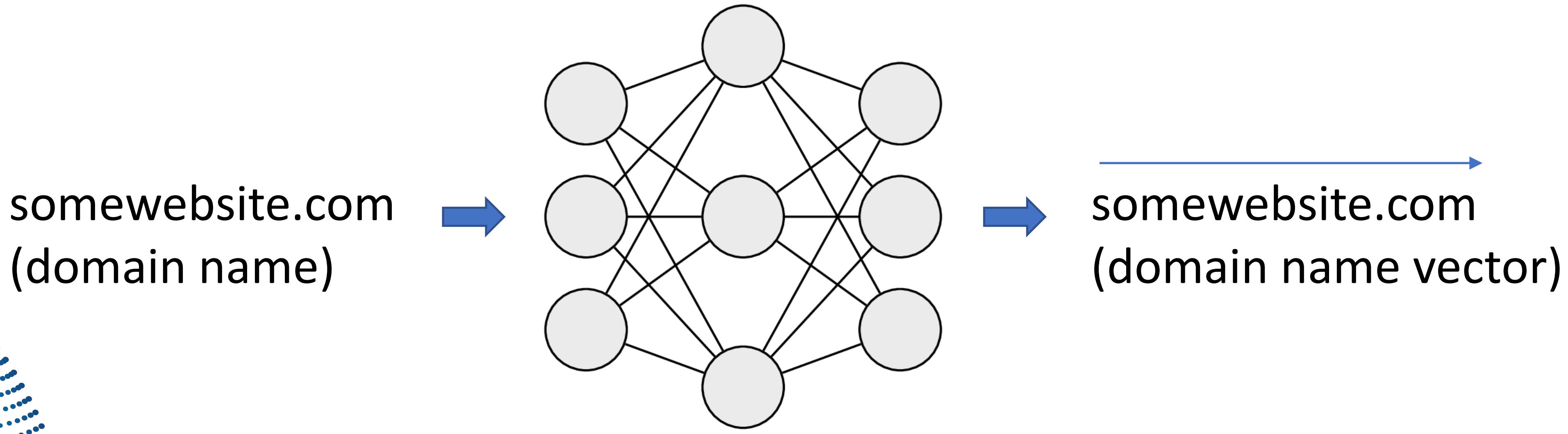
Problem Statement

- Network traffic is the same, we want to prioritize, shape, filter different types of traffic BUT..
  - Everything (well almost, everything) is HTTPS
  - We cannot see content length or other indicators in the header (HTTPS) so cannot estimate the duration of connection
  - We do still know the destination in terms of the Domain Name and IP address.



Roadmap to a potential solution

- Given that we know the domain name (destination) what can we say about the traffic?
- To be able to do anything useful with the domain name which is qualitative/categorical we need to convert it to quantitative/numerical.



somewebsite.com
(domain name)

$\overrightarrow{\text{somewebsite.com}}$
(domain name vector)

Roadmap to a potential solution

- Collect DNS traffic packet trace from an ISP’s core network.
- Inspired from Word2Vec and StarSpace, wrangle the data to a format which can be used to train a Word2Vec “skipgram” model.
- The output is a vectorized representation of a domain name and this vector space representation now unlocks possibilities for a whole host of traffic engineering applications.

Implementation Details – Data Collection

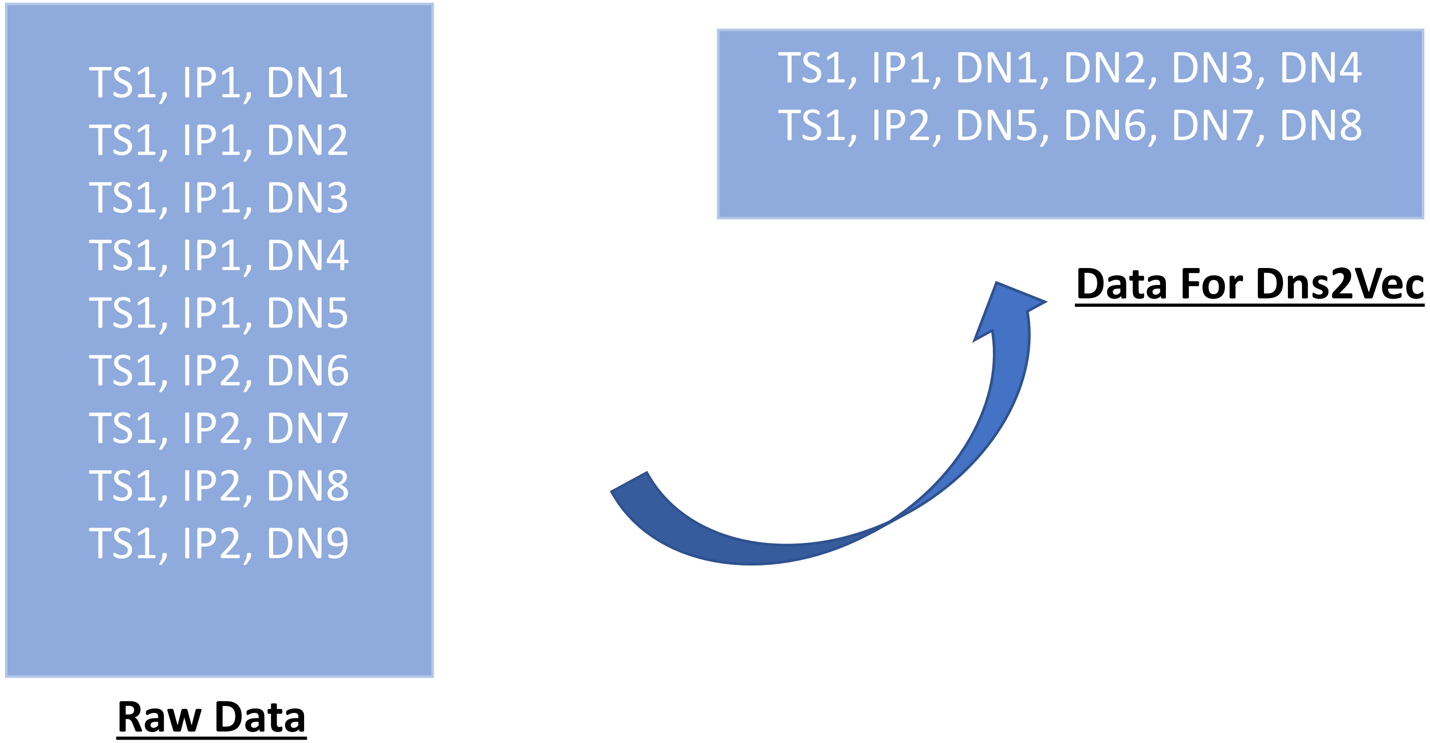
- DNS traffic was collected from a large ISP for a 3-day period.
- The dataset consisted of CSV files (one for each day) in the format *<timestamp>, <Destination IP address>, <Source IP address>,<Query Name>., <Query Type>*

```
Oct 22, 2018 02:29:38.680373000 202.95.128.180  10.128.17.185   config-api.internet.apps.samsung.com          0x0001
Oct 22, 2018 02:29:38.761780000 202.182.182.182 10.128.146.105  googleads.g.doubleclick.net           0x001c
Oct 22, 2018 02:29:38.768370000 202.95.128.180  10.128.17.185   config-api.internet.apps.samsung.com          0x001c
Oct 22, 2018 02:29:38.970772000 202.95.128.180  10.128.17.185   api.foursquare.com            0x001c
Oct 22, 2018 02:29:39.053499000 8.8.8.8 10.128.19.124   c.disquscdn.com        0x0001
```

- The files were combined into a single file which had a total of 589 million rows (one DNS query per row) and provided to a data cleaning and wrangling module.

Implementation Details – Preparing For Dns2Vec

- To prepare data for Dns2Vec we group all DNS lookups from an IP address in a 20-minute period and that constitutes a “Document” with each individual Domain Name is a “Word”.



TS1, IP1, DN1
TS1, IP1, DN2
TS1, IP1, DN3
TS1, IP1, DN4
TS1, IP1, DN5
TS1, IP2, DN6
TS1, IP2, DN7
TS1, IP2, DN8
TS1, IP2, DN9

Raw Data

TS1, IP1, DN1, DN2, DN3, DN4
TS1, IP2, DN5, DN6, DN7, DN8

Data For Dns2Vec

Implementation Details – Data Wrangling

- To remove the heterogeneity from domain names belonging to the same entity the domains were shortened to 2 or 3 significant levels (for domains ending in country specific suffixes) resulting in 1.05 million unique domains.
  - For example, a.yahoo.com and b.yahoo.com both get reduced to yahoo.com.
- The file was parsed, each line tokenized and a new timestamp aggregation field was created to group timestamps into 20-minute intervals.
  - The idea being that we want to capture all DNS lookups from the same browsing session so the source IP address and a short time window provide a convenient way to do this.

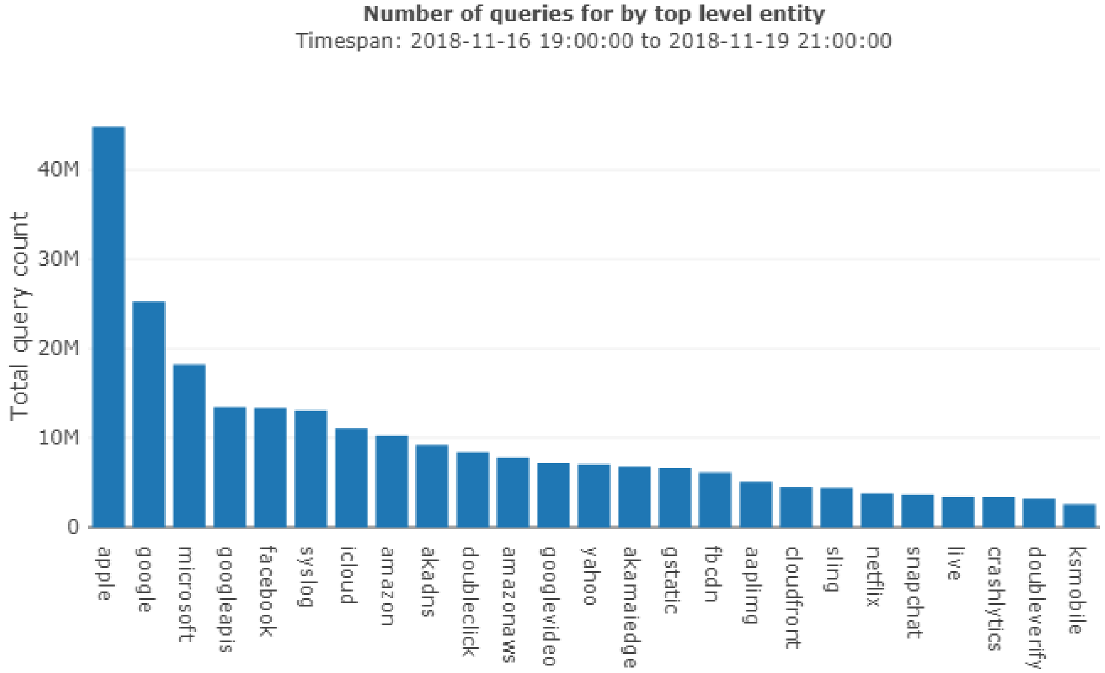
Implementation Details – Data Wrangling

- This is what the wrangled data ready for Word2Vec looks like:

```
ts_aggr, subnet, query
201809271201,100.64.192.192,update.googleapis.com

201809271201,100.64.192.232,cs.dds.microsoft.com telecommand.telemetry.microsoft.com buddy.bitdefender.com client-office365-tas.msedge.net ecn.dev.virtualearth.net clientservices.googleapis.com spclient.wg.spotify.com cdn.content.prod.cms.msn.com vmeasureul.dishaccess.tv activity.windows.com elb-nvi-amz.nimbus.bitdefender.net api.login.yahoo.com

201809271201,100.64.194.88,xtra3.gpsonextra.net xtra1.gpsonextra.net dc3-vault.myvzw.com events.appsflyer.com xtra1.gpsonextra.net tpc.googlesyndication.com

201809271201,100.64.195.248,calendar.google.com gsa.apple.com

201809271201,100.64.195.80,qwerty1.lorexddns.net iphonesubmissions.apple.com www.wdc.com d3ot8v94c25lje.cloudfront.net www.wdc.com 6-courier.push.apple.com qwerty1.lorexddns.net eco3dba7b5b1.282.ozvision.ozsn.net qwerty1.lorexddns.net ddns.lorexddns.net qwerty1.lorexddns.net
```

Implementation Details – Data Cleaning

- We are not ready just yet, there are some domains that are just everywhere, repeated constantly, they need to be removed.
- Instead of just a frequency count based removal, we use TF-IDF (Term Frequency, Inverse Document Frequency) score to remove “Stop Domains”.



Implementation Details – Data Cleaning

- Instead of a simple frequency count based removal, we use TF-IDF (Term Frequency, Inverse Document Frequency) score to remove “Stop Domains”.

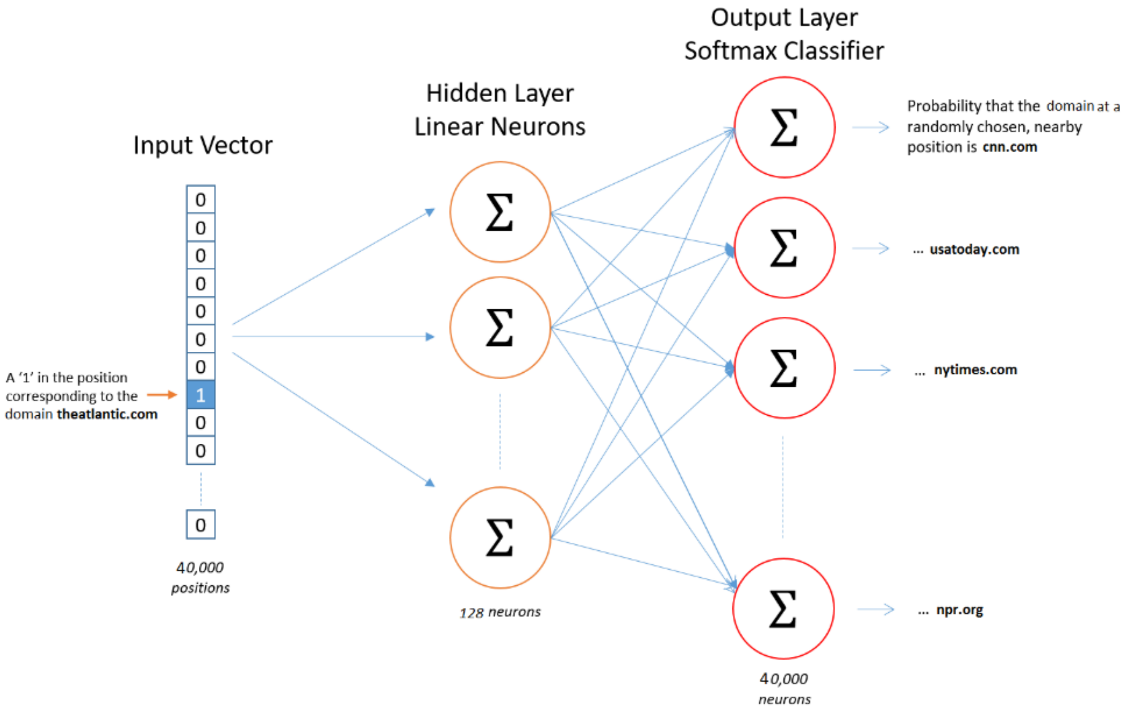
Top 10 stop domains based on TF-IDF
www.google.com www.apple.com apple.com www.icloud.com myip.sling.com time-ios.apple.com www.facebook.com imap.gmail.com msg.sling.com mesu.apple.com

Implementation Details – Word2Vec

- The corpus thus created was fed to the Gensim package’s Word2Vec implementation.
- Hyperparameters of the model:

Hyperparameter	Chosen Value	Explanation
Vocabulary Size	40,000	Number of domain names chosen for creating the embeddings
Sample	10^{-5}	The threshold for configuring which higher-frequency words are randomly downsampled
Negative Exponent	-1	The exponent used to shape the negative sampling distribution. A value of 1.0 samples exactly in proportion to the frequencies, 0.0 samples all words equally, while a negative value samples low-frequency words more than high-frequency words.
Embedding Size	128	Dimensionality of the embedding vector
Window	7	Maximum distance between the current and predicted word within a sentence
Negative Samples	5	How many “noise words” should be drawn (usually between 5-20)
Training Iterations	150	Number of iterations (epochs) over the corpus

Implementation Details – Word2Vec



Input Vector

A '1' in the position corresponding to the domain **theatlantic.com**

40,000 positions

Hidden Layer Linear Neurons

128 neurons

Output Layer Softmax Classifier

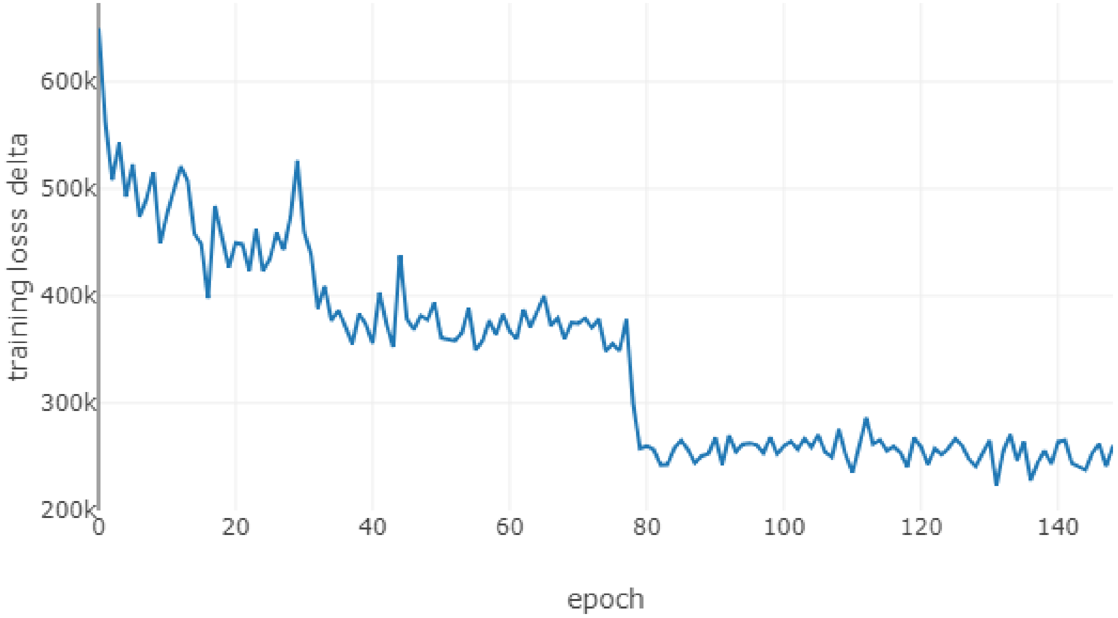
40,000 neurons

Probability that the domain at a randomly chosen, nearby position is **cnn.com**

... **usatoday.com**

... **nytimes.com**

... **npr.org**



Results – Similar Domains

- Results were examined using the following three methods
  - **Method 1**: Finding similar domain names to common domain names in different categories.
  - This translates into using Cosine Similarity to find similarity between vectors for the domain of interest and the rest of the domains (words) in the dataset (vocabulary).

theatlantic.com	nytimes.com, getpocket.com, npr.org, elle.com
food.com	geniuskitchen.com, whisk.com, addapinch.com, foodnetwork.com
glassdoor.com	ziprecruiter.com, indeed.com, glassdoor.de

Results – Domain Analogies

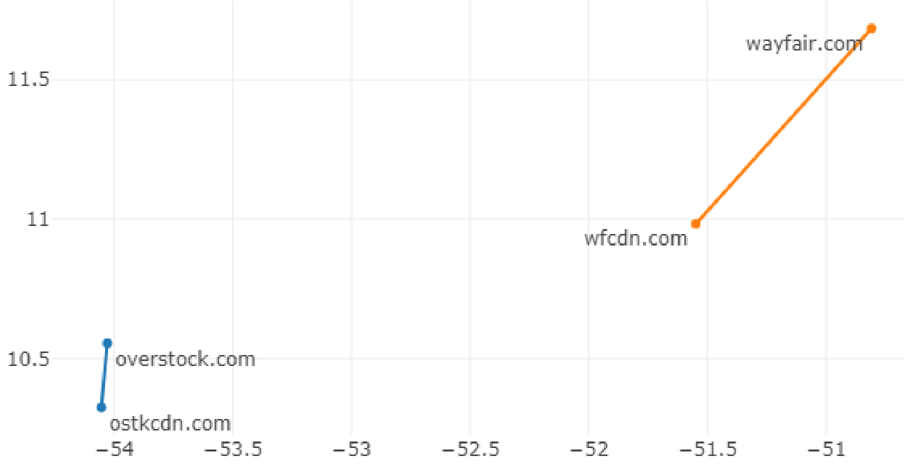
- No fun in doing Dns2Vec (Word2Vec) without word (domain) analogies

wfcdn.com - wayfair.com + overstock.com = ovstkcdn.com (find CDN for a website)

delta.com + hilton.com = tripadvisor.com (airline and hotel combine to form a trip)

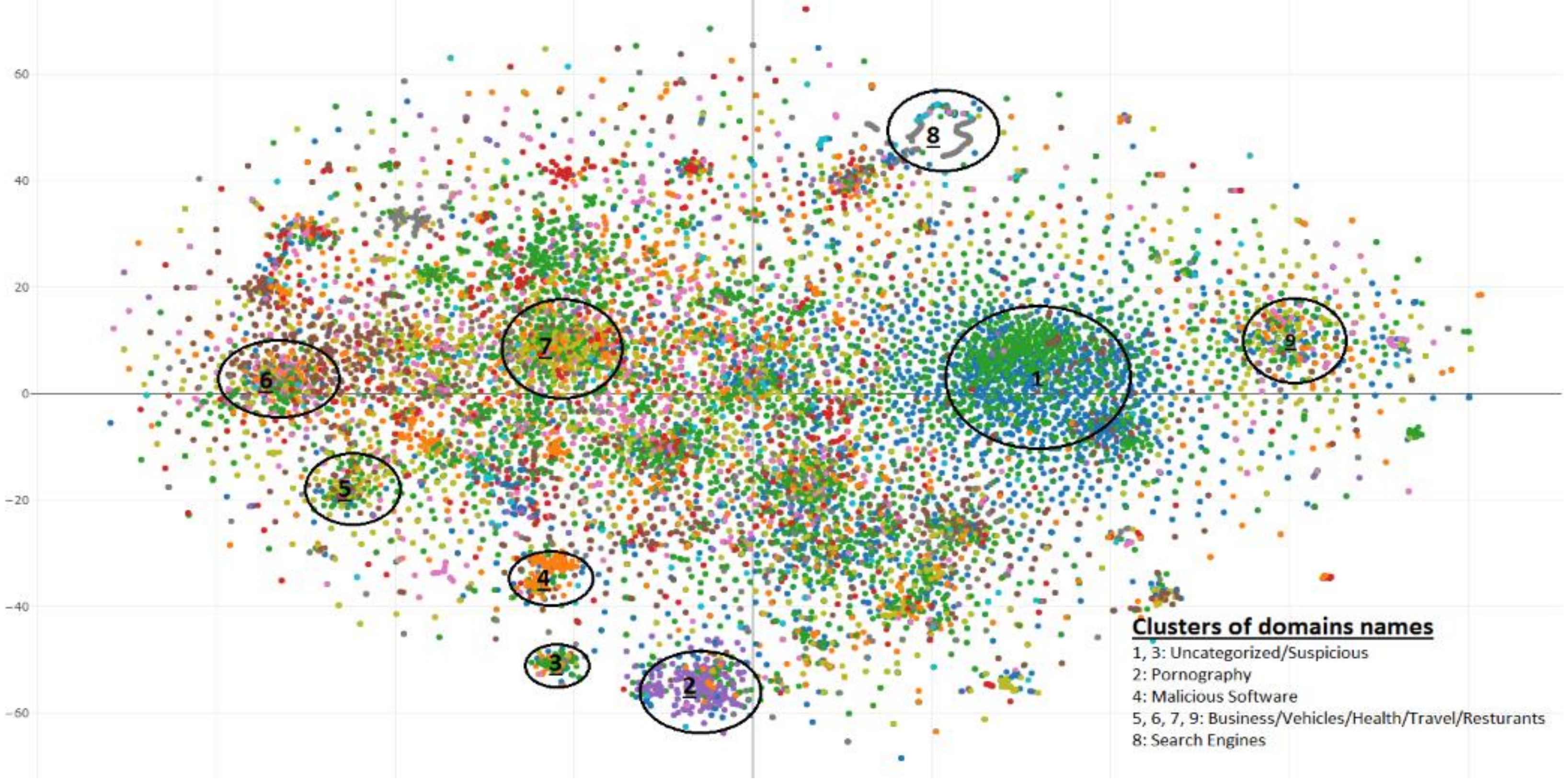
Domain name analogies with Dns2Vec

wfdn.com (wayfair content delivery network) is to wayfair.com
as ostkcdn.com (overstock content delivery network) is to overstock.com



Results – Category clusters

- **Method 2**: Assign a category such as “search engine”, “sports/entertainment”, “adult” to each domain name using the Symantec K9 service and then visualize the clusters using t-SNE.



Results – Category clusters

- **Method 3**: A simple quantitative variation of method 2.
- We find top 3 most similar domains to every domain in the dataset and then count a “match” if at least 1 of the top 3 similar domains belongs to the same category (as determined from Symantec K9) as the input domain name.
- A better overall score means better quality domain vectors.
- This technique was used for determining the best set of hyper parameters for training Word2Vec.
- The best accuracy as measured by this technique was 55%.

Discussion – How DNS vectors help in traffic engineering

Categorizing new domain names in various groups i.e. news media, sports/entertainment, adult, malware, online shopping, API etc.

Prioritizing traffic based on domain name categorization right from the first packet onwards

Use as an additional feature in ML models for identifying malicious traffic

Use DNS vectors from browsing session to predict next set of domains (Seq2Seq)

Use DNS vectors from browsing session to create a browsing session vector

Thank you!

Questions?