IntelliSearch: A novel and personalized approach to domain name curation

Steven Tey

Minerva Schools at KGI
B.Sc. Computational Sciences
B.Sc. Brand Management

Abstract

In this paper, we present a novel approach to domain name curation by using deep learning and natural language processing (NLP) with the help of fastText word vectors, word embeddings, RESTful APIs, neural networks, Markov chains, and various other machine learning algorithms. We also use collaborative filtering, nearest neighbor search, and KD-trees to make the curated recommendations more personalized over time.

1 Background & Context

On March 15, 1985, the domain name 'symbolics.com' was registered by a computer manufacturer in Cambridge, Massachusetts, essentially making it the first-ever .com domain to be registered in history. 35 years later, over 350 million domain names have been registered [1], and yet the process of domain name registration has stayed fairly similar throughout the last 3 and a half decades - you think of a good name, type it into your registrar of choice, check if it's available, and if it is, you register it. Sure, there are a few domain name generators like Lean Domain Search or Instant Domain Search out there, but their recommendations are usually very deterministic. For example, if you type in 'data' as your search query, you will receive results like 'datasrus', 'datacosmetics' or 'datasforsale', which are not very creative or helpful whatsoever, and that's because these sites usually just add a bunch of different words as prefixes or suffixes to your search query and call it a day.

Without a doubt, there has been a dearth of innovation in the domain name industry. Sure, it is not the hottest or sexiest industry in Silicon Valley, but it ultimately forms the basis of the

branding industry - an industry where companies spend up to $600 billion dollars a year building brand value for their corporate entities [2]. Take branding agencies like Lexicon Branding for example - by focusing solely on naming and brand architecture, they have managed to make a reported $60 million dollars in annual revenue [3]. Furthermore, Lexicon is just one of the many branding agencies in the world, which makes this an underrated, yet very lucrative business space.

Of course, naming the next billion dollar company is definitely not as simple as adding suffixes and prefixes to the industry that the company operates in - it requires a lot of creativity, consumer research, linguistic analysis, trademark evaluations, and many more. However, while being a complex, multi-faceted business, it is one that is ripe for disruption, and I believe that there is a lot of potential for creative machine learning applications in this sector. Therefore, this paper will focus on the implementation of various deep learning algorithms with the goal of finding the perfect recipe for intelligent and personalized domain name curation.

2 Overview

The goal of this paper is to produce a novel domain curation algorithm, *IntelliSearch*, as the final deliverable. *Intellisearch* consists of three components:

- **Domain Name Synthesizer**: The creative component of *IntelliSearch* that uses Torch recurrent neural networks (RNN) and Markov chains that are trained on millions of existing startup names to come up with similar-sounding ones. These startup names are also tagged with the industry(ies) that they are suitable for and the tags are taken into account when these names are being generated to make sure that the generated names are automatically categorized into their respective industries.
- **Domain Name Scraper**: The self-driving web spider component of *IntelliSearch* that scrapes the internet using various RESTful APIs to find the best available domain names that are currently available for registration/purchase via various domain aftermarket and auction platforms.
- **Domain Name Curation**: The AI component of *IntelliSearch*, which uses fastText word vectors, nearest-neighbor search, KD-trees, and collaborative filtering to help sort through the millions of domain names produced by the previous two components and categorize them based on the users' preferences.

3 Methodology

The build process of this algorithm can be broken down into 3 main steps:

1. Preprocessing raw data: All the raw data collected by the Domain Name Scraper will be cleaned using methods like lemmatization, stemming, and deduplication.

2. Name generation: The Domain Name Synthesizer will analyze tens of thousands of data points consisting of startup names from various different industries and come up with an exhaustive database of sample names for each of these categories.
3. Feature engineering: Typeform surveys will be sent out to a focus group of 50 beta testers with a list of different questions to help identify their individual preferences. Examples of those criteria are as follows:
    a. Industry preference (fintech, healthcare, EdTech, etc.)
    b. Word length (number of characters)
    c. Languages (English, Latin, Greek, etc.)
    d. Word categories (portmanteaus, word combinations, misspellings, etc.)
    e. Preferred registrar (GoDaddy, Namecheap, Epik, etc.)
    f. Budget
    g. TLDs
    h. Personal interests (hunting, running, archery, etc.)
    i. Reg-free domains only? (no Aftermarket/Auction Domains)
    j. Other criteria (SEO ranking, backlinks, number of registered TLDs, etc.)
4. Gather all the information and build a classification model tailored to each user's needs. The results produced by the model will then be ranked by their cosine similarity scores, semantic similarities using nearest neighbor lookups, Levenshtein distance to dictionary words [4], Scrabble score, and various other factors that determine brandability and pronounceability.

4 Intended Deliverables

Here's a breakdown of the type of names that the Domain Name Synthesizer will be producing:

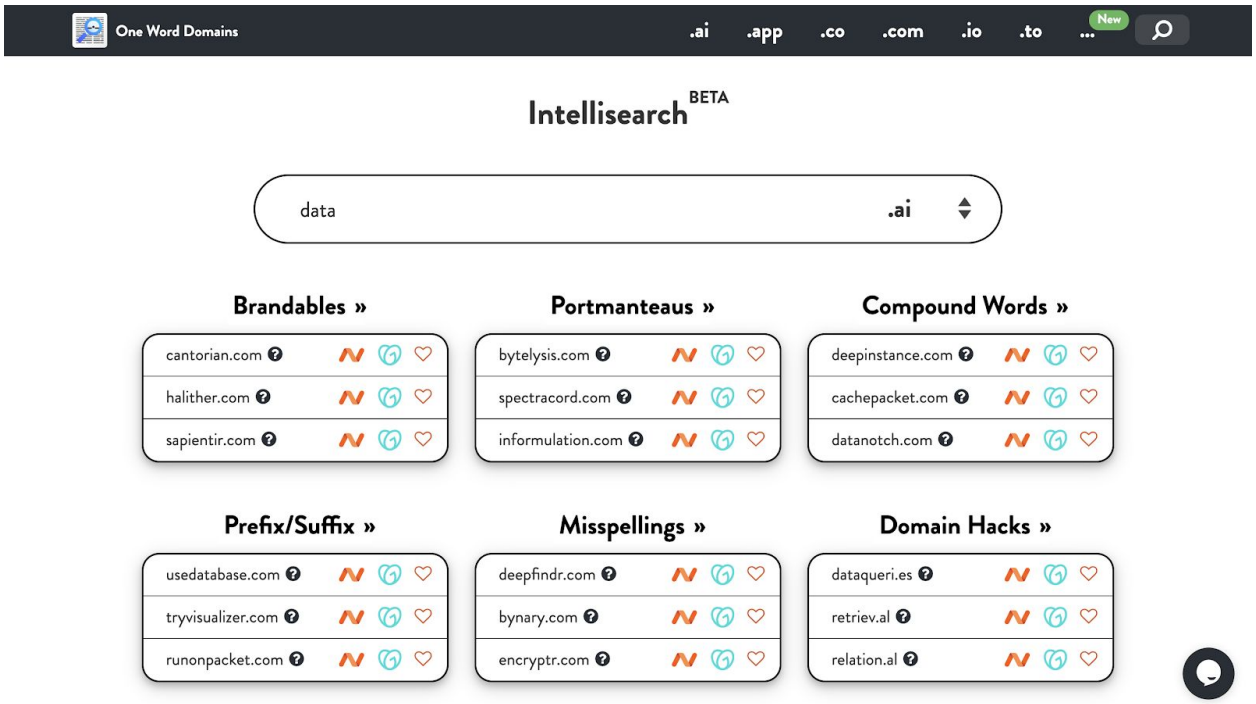
- Basic (Root Words):
    - Adjectives (superlatives included): Earnest, Jolly, Brightest
    - Verbs: Ride, Cruise, Deserve
    - Nouns: Stilt, Figure, Root
    - Adverb: Smartly, Musically, Cleverly
- Advanced:
    - Word associations: E.g. for a given term 'data', associated words could be 'bytes', 'packet', and 'visualize'.
    - Portmanteaus: Accenture, Vercel, Finimize
    - Word combinations: Glassdoor, Robinhood, Facebook
    - Misspellings: Flickr, Disqus, Tumblr
    - Brandable Modifiers: Shopify, Lendable,
    - Other languages: Kairos, Calypso, Kuaishou
    - Play on the names of famous people: Knewton, Eulerity, Socratic
    - Fictional/Made-up Names: Palantir

And here's a list of the different startup industries that will be taken into consideration by the *Intellisearch* model (note: this list is not exhaustive, at least not yet):

- FinTech: Robinhood, PayPal, Brex
- BioTech: Pfizer, Freenome, Novartis
- EdTech: Coursera, Udemy, Knewton
- Productivity SaaS: Notion, Slack, Basecamp
- Video Conferencing SaaS: Zoom, Bluejeans, Hangouts
- HR SaaS: Workday, Glassdoor, Greenhouse
- CRM SaaS : Salesforce, Zendesk, Hubspot
- API SaaS: RapidApi, Zapier, Twillio
- Email Tools: Mailchimp, Sendinblue, ConvertKit
- File-sharing: Dropbox, Box, OneDrive
- Social Media: Facebook, Instagram, Twitter
- Big Data: Terracotta, Wavefront, Prognostic
- AI/ML: DataRobot, ScaleAI, Deepnote
- Autonomous Vehicles: Cruise, Pony, Embark
- Web Hosting: Heroku, Vultr, DigitalOcean

5 Demo

Here's a Loom video that shows a proof of concept of the final product and how it's supposed to work:



Appendix A

By the end of the summer, I should have a well thought out framework for how I will be building the *Intellisearch* model, which also includes having a robust training dataset that consists of the following:

- An exhaustive list of the different startup industries that exist in the world - ranging all the way from software to hardware and even to construction and fashion.
- A list of the top 10,000 startup names that are tagged with their respective industries (a startup can also span across multiple industries given the granularity of the dataset that is being taken into account in the previous step).

By the end of the fall semester, I should have a working prototype of the deep learning model that can perform the following tasks:

- Allow people to enter a search query and display the appropriate search results (as shown in the Loom screen recording in item 5).
- Perform minimal reinforcement learning and collaborative filtering by taking into account feedback from the user to help improve search & curation performance.
- Tested the prototype with at least 50 beta testers, ideally more, to get feedback.

By the end of the spring semester, I should have a fully-functioning product that is powerful, personalized, and scalable. Here are some nice-to-have goals to accompany that:

- Have a feature for users to input their business idea in natural language and use that to general names that are appropriate for their use case.
- If the previous step is too ambitious, maybe make it more containerized and structured by allowing users to describe various attributes of their idea using multiple-choice questions.
- Start piloting the product with paying customers and constantly iterating and improving it over time.

Appendix B

Applicable HCs:

- #algorithms
- #optimization
- #designthinking
- #purpose
- #context
- #interpretivelens

- #hypothesisdevelopment
- #audience
- #rightproblem
- #breakitdown
- #variables

Applicable LOs

- #cs156-neuralnetworks
- #cs156-modelmetrics
- #cs156-classification
- #cs112-decisiondata
- #cs112-decisioninference
- #b154-brandConnection
- #b144-ideation

References

[1] BusinessWire. (2019). *Internet Grows to 354.7 Million Domain Name Registrations in the Second Quarter of 2019.* Retrieved from https://www.businesswire.com/news/home/20190829005729/en/

[2] ANA. (2016). *Marketing Spend on Brand Activation will top $595 Billion in 2016: ANA Report.* Retrieved from https://www.ana.net/content/show/id/39647

[3] RocketReach. (2020). *Lexicon Branding Information.* Retrieved from https://rocketreach.co/lexicon-branding-profile_b5c6c3acf42e0cce

[4] Wikipedia. (2020). *Levenshtein Distance.* Retrieved from https://en.wikipedia.org/wiki/Levenshtein_distance