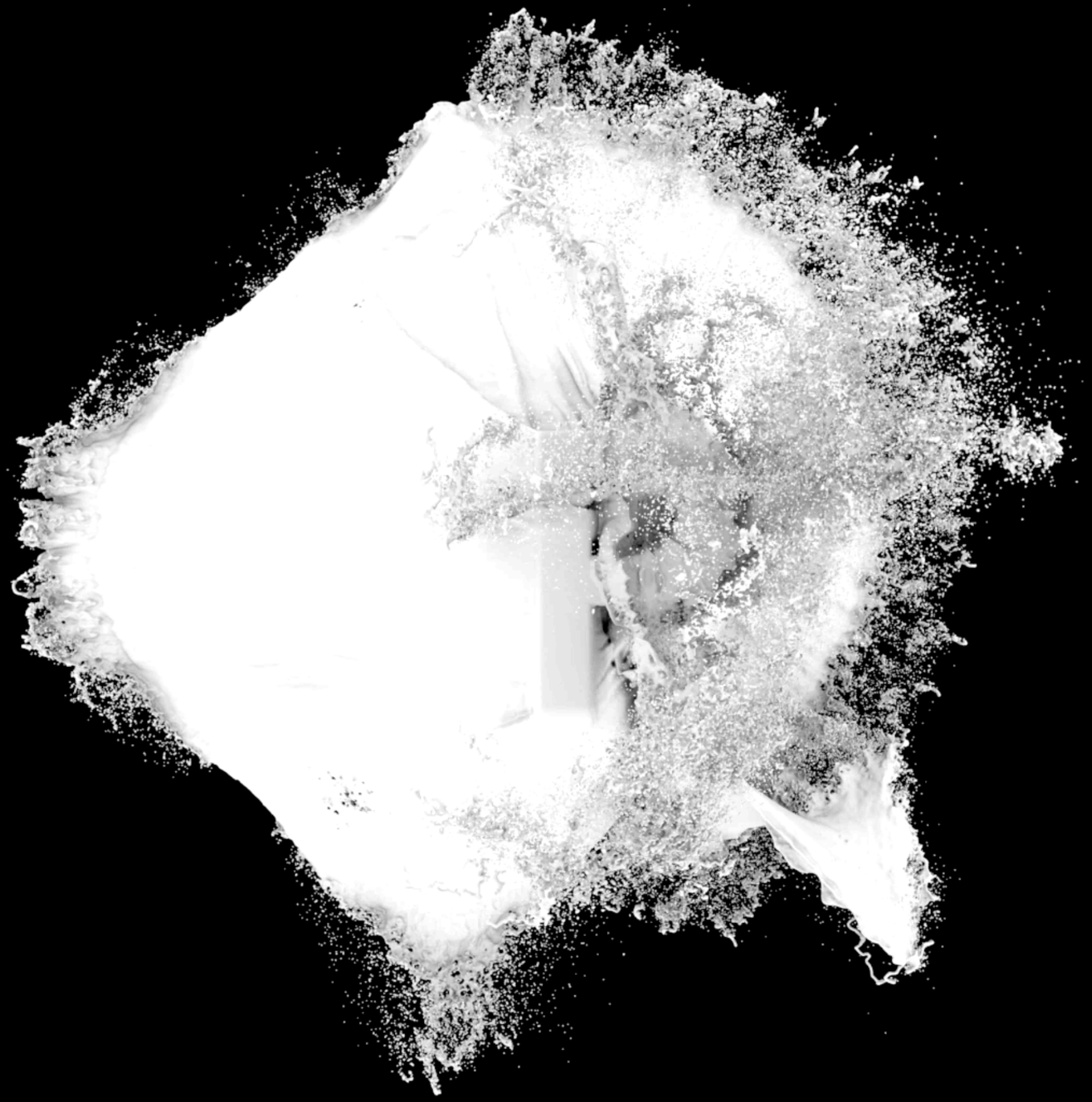


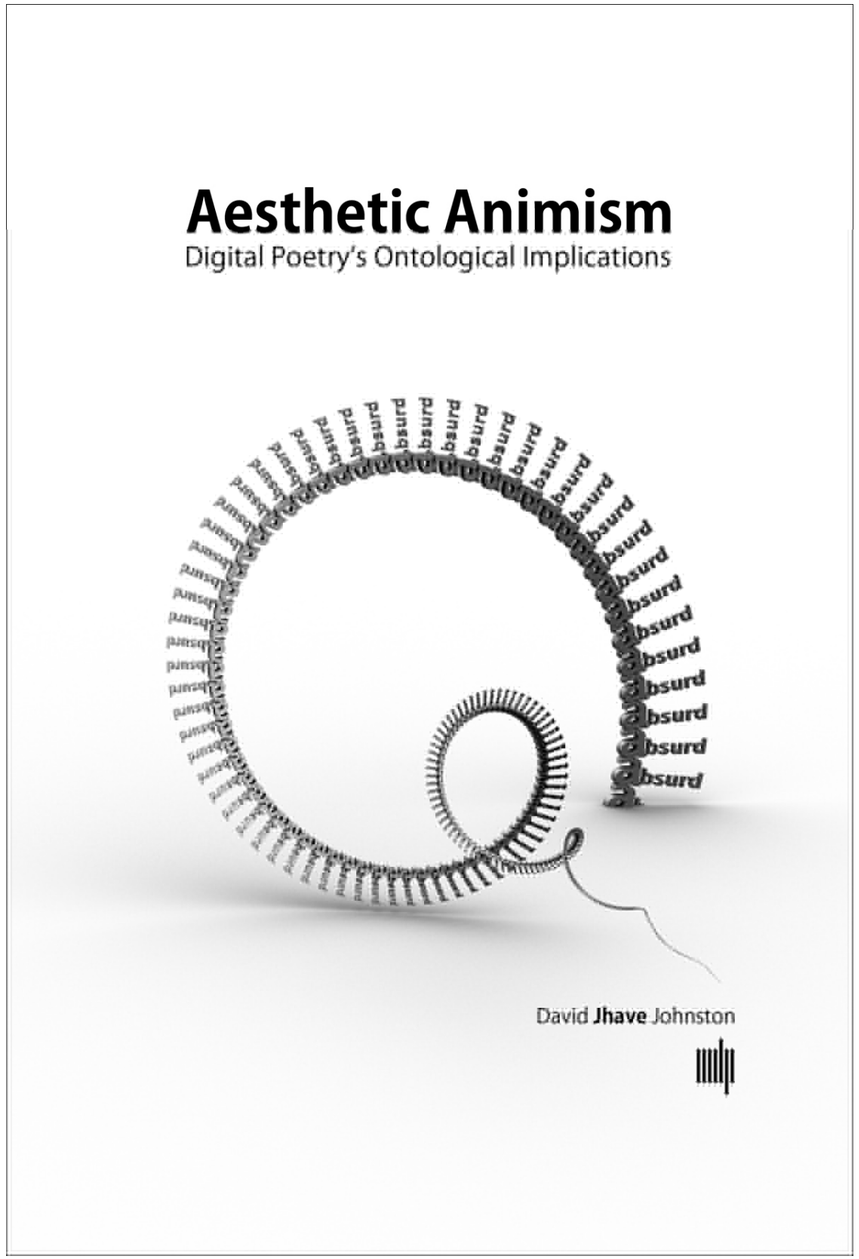
Writing
with A.I. and Machine Learning

David (Jhave) Johnston
glia.ca

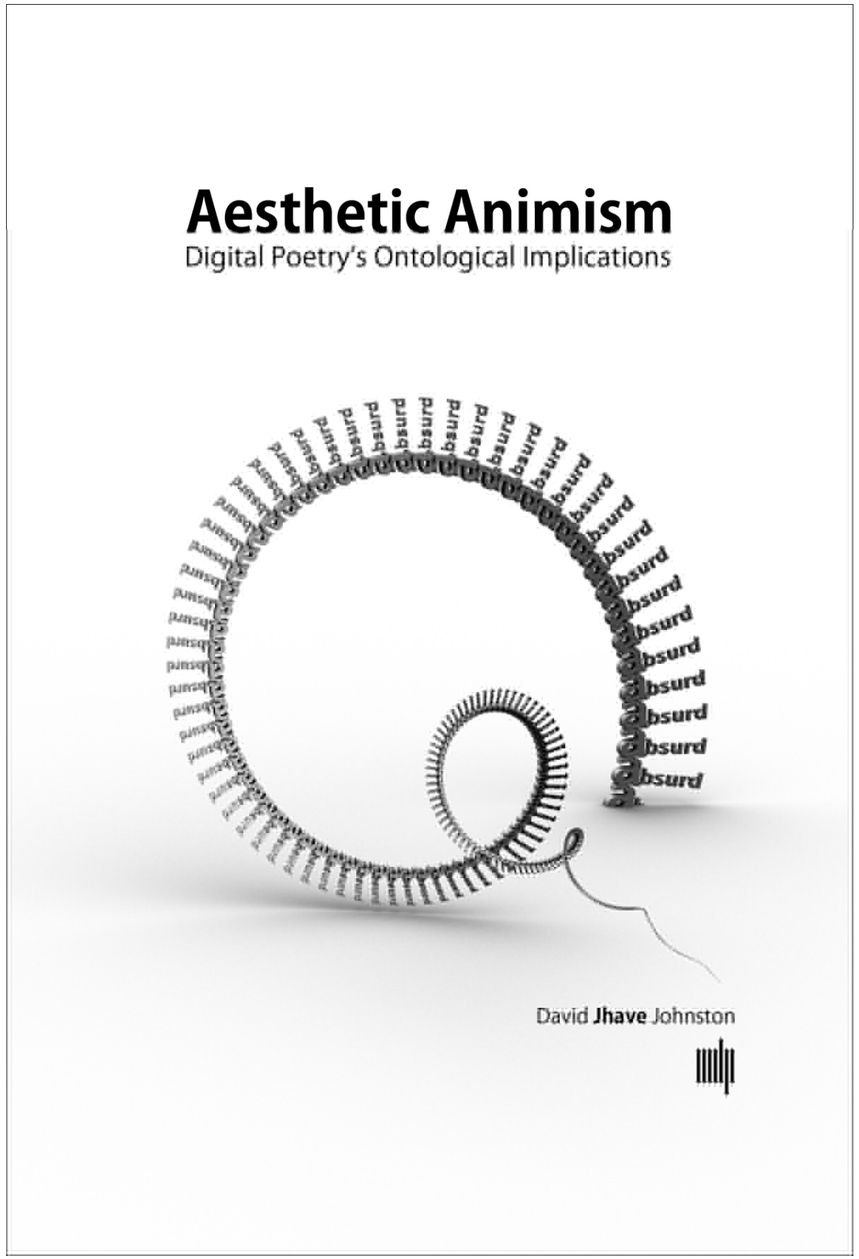
when they opened me as i believed,
i thought i'd found
the face

of a new and intelligent
artifice, a river
afraid of the earth

and the jaundiced eyes
of the night, and the air
far out at sea



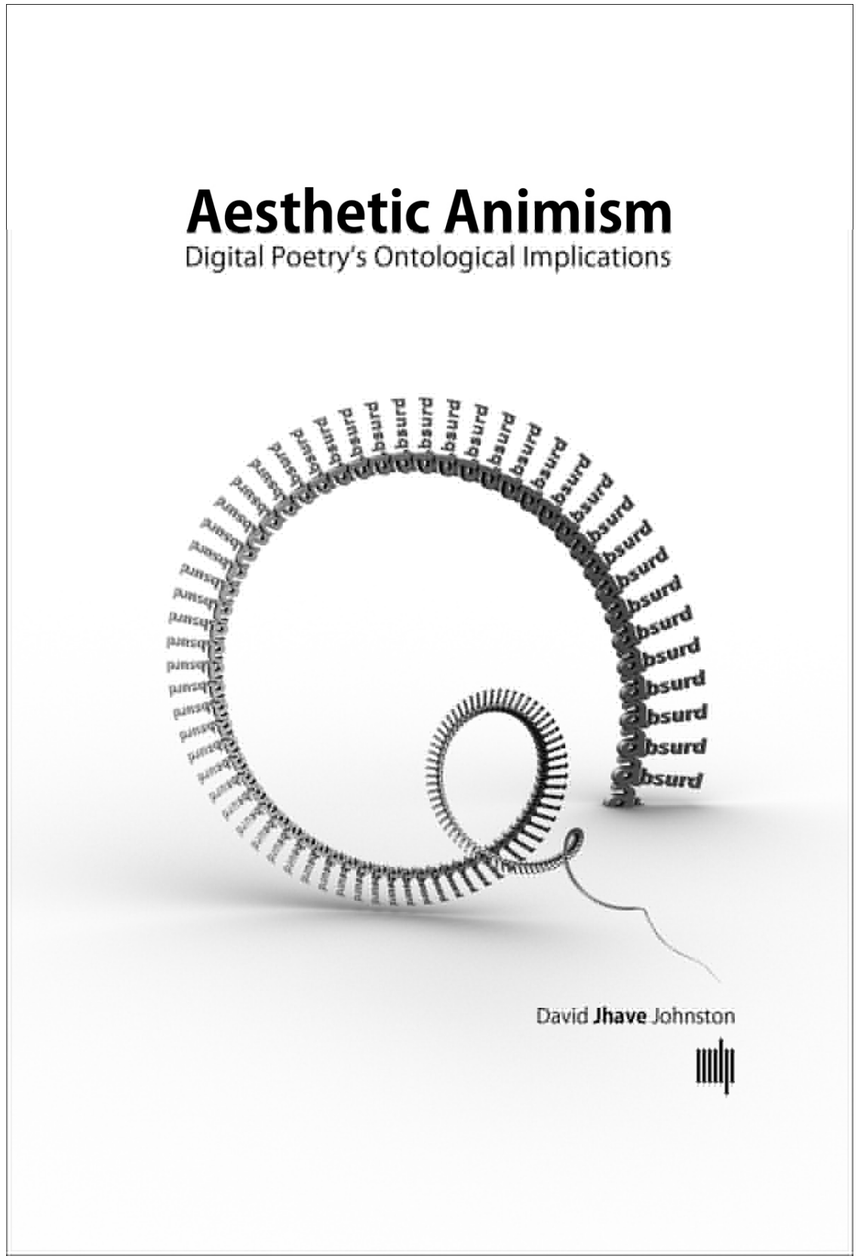
This book offers a decoder for some of the new forms of poetry enabled by digital technology.



Digital poems can be ads, conceptual art, interactive displays, performative projects, games, or apps.

Poetic tools include algorithms, browsers, social media, and data.

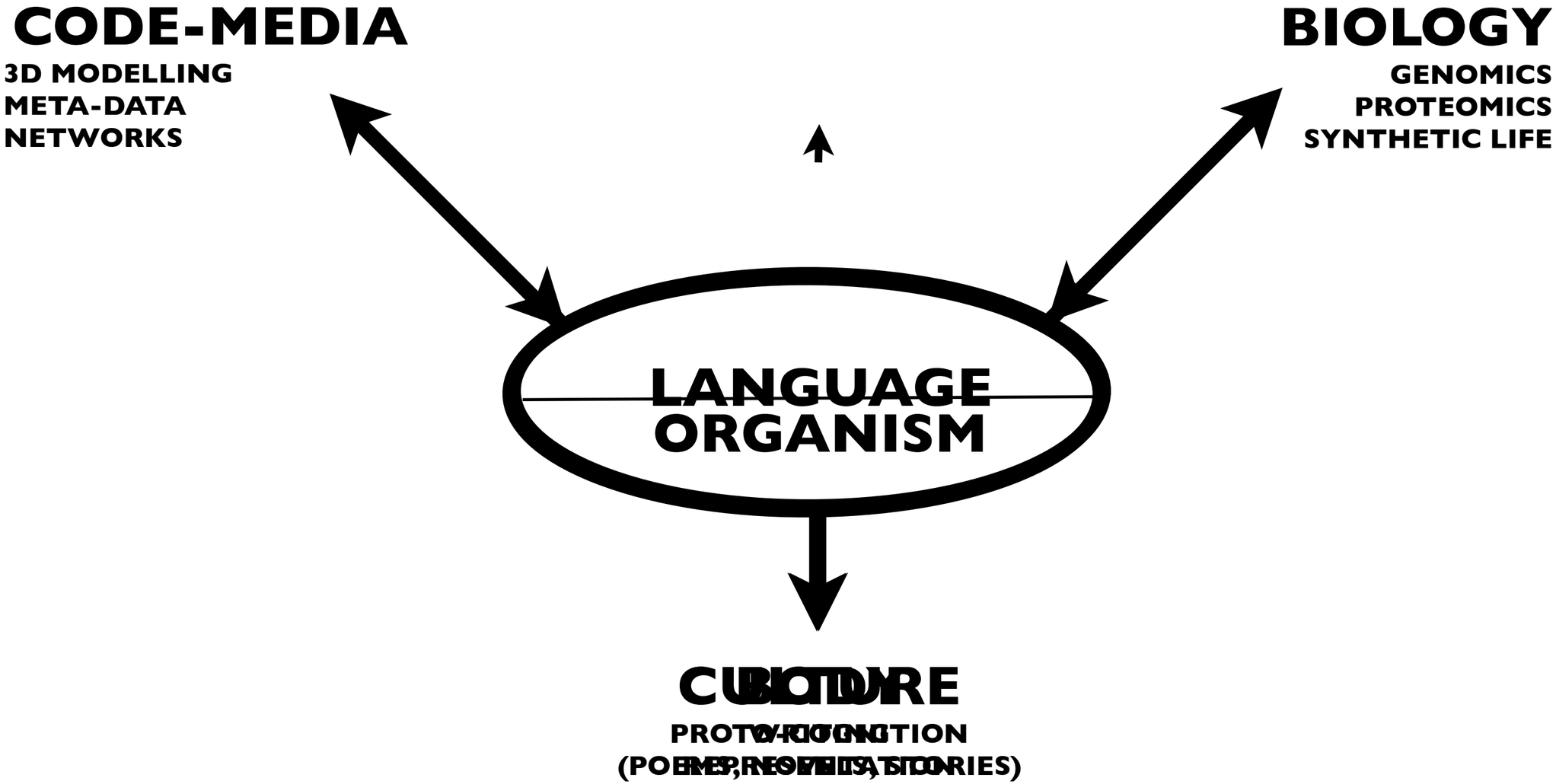
Code blossoms into poetic objects and poetic proto-organisms.



In the future imagined here, digital poets program, sculpt, and nourish immense immersive interfaces of semi-autonomous word ecosystems.

Poetry, enhanced by code and animated by sensors, reengages themes active at the origin of poetry: animism, agency, consciousness.

I am an **artist** taking refuge in **academia**.


CODE-MEDIA
3D MODELLING
META-DATA
NETWORKS
BIOLOGY
GENOMICS
PROTEOMICS
SYNTHETIC LIFE
LANGUAGE
ORGANISM
CULTURE
PROTOCOLS
(POEMS, NOVELS, STORIES)

META

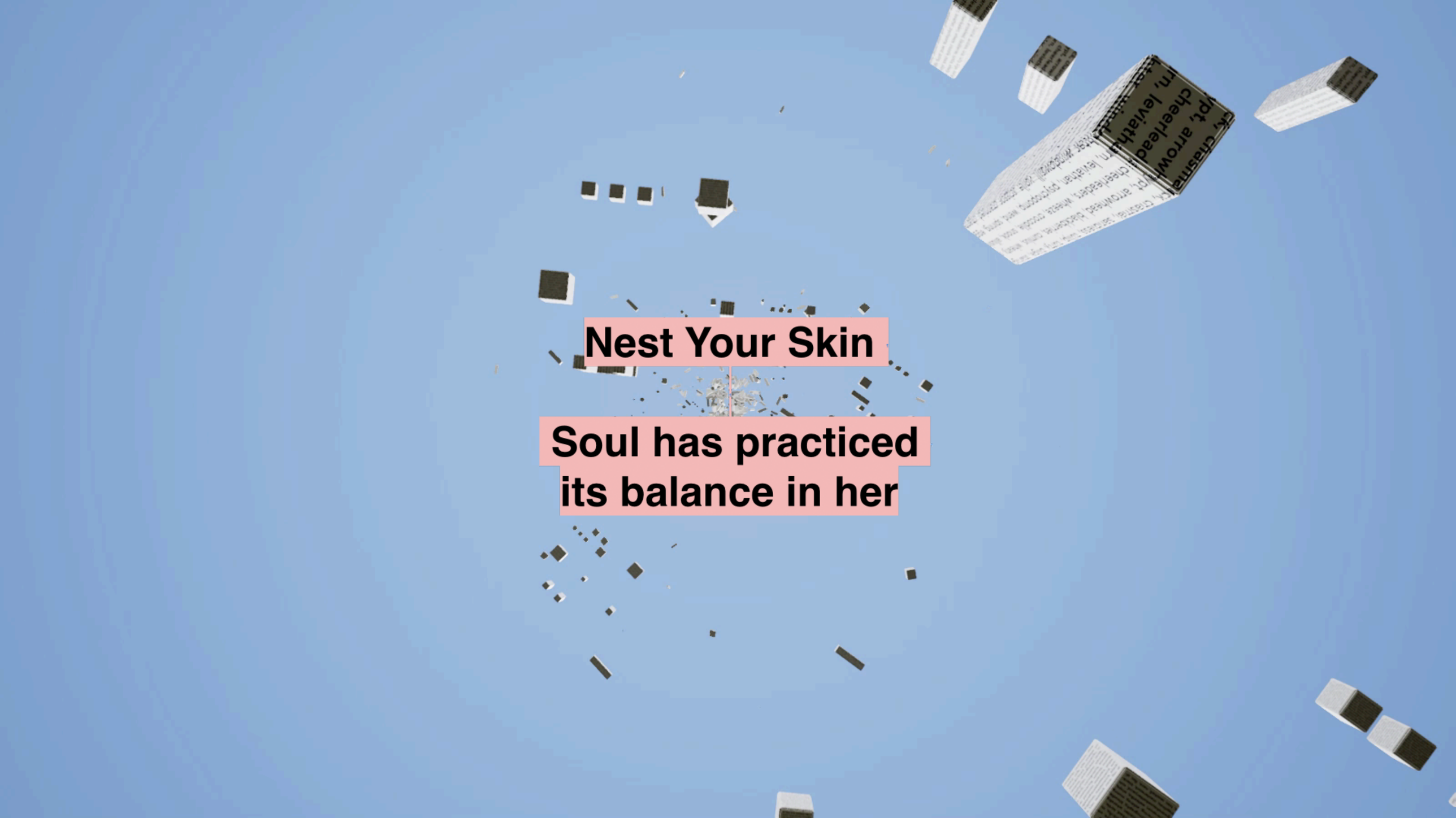
PORE

The poem fakes
And fakes so well,
It manages to fake
Pain really felt

And those who read
Feel clear pains:
Un-intended,
Un-sensed.

And thus, jolting on its track,
Busy reason,
Circling like a clock
Calls itself a heart.

Fernando Pessoa, *Autopsychography*


Nest Your Skin
Soul has practiced
its balance in her

Generative Adversarial Algorithms

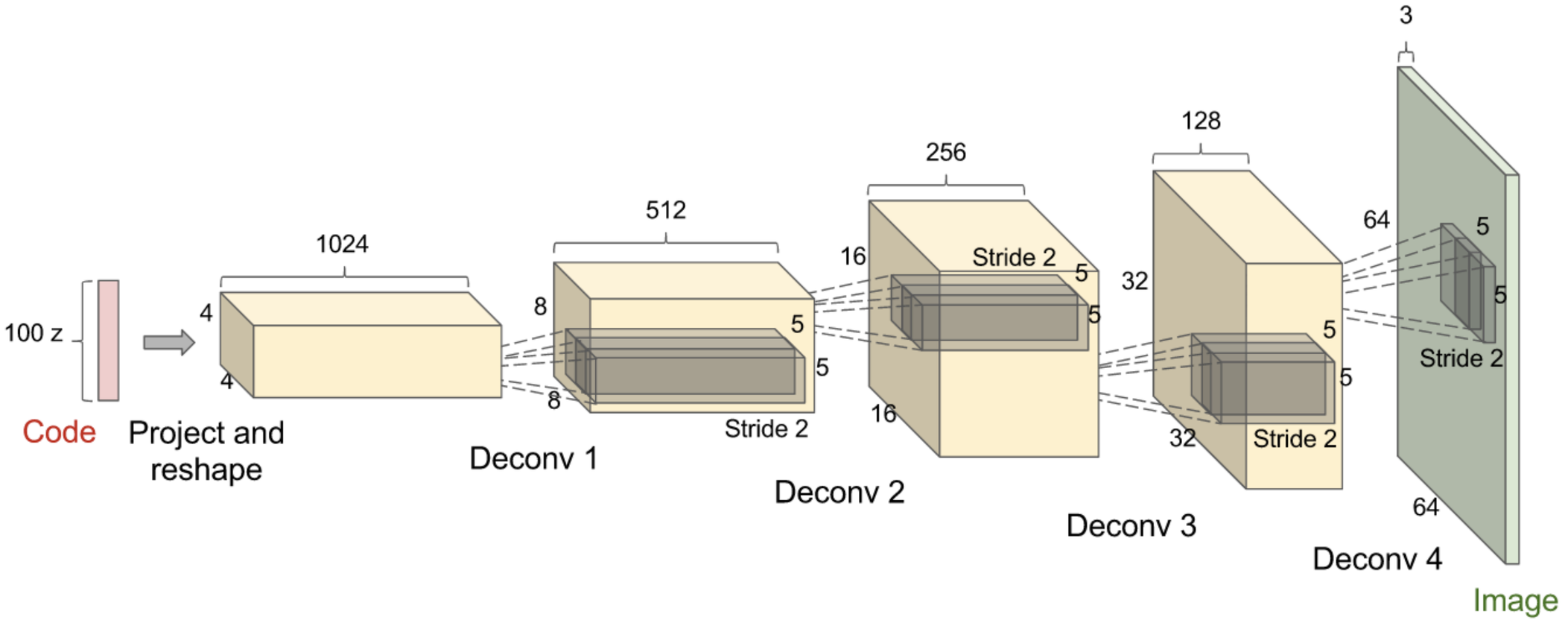
are *neural networks* that belong to a branch of *unsupervised learning.*

Goodfellow, Ian J.; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). "Generative Adversarial Networks". arXiv:1406.266

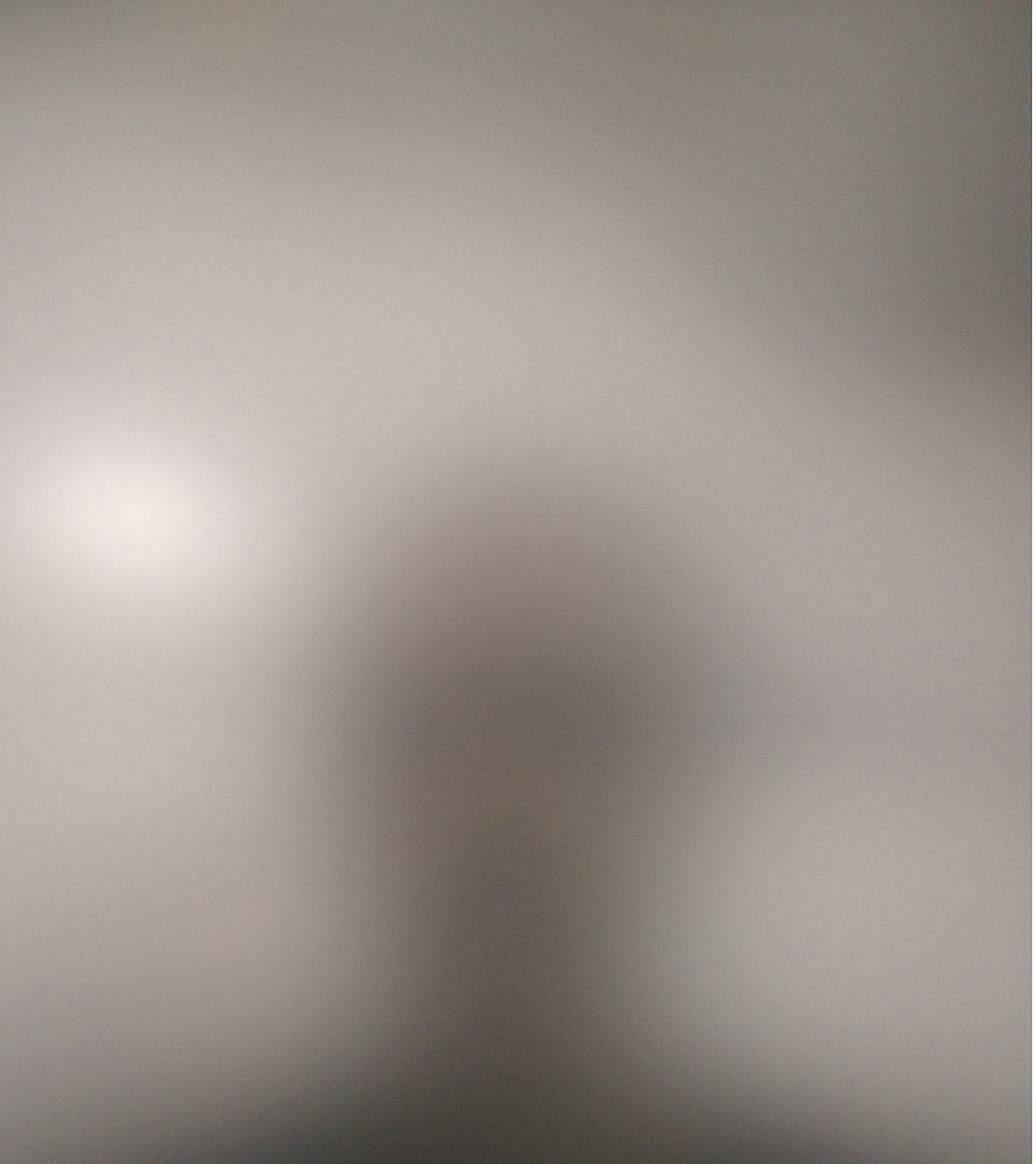
Think of a **neural net** as a mathematical approximation of a **brain**. Its brain begins empty, it is a newborn **baby**.

Consider how a baby learns how to speak its first words: it is not told explicitly about syntax, grammar.

It listens.



In **unsupervised learning**, an algorithm is fed (trained on) *unlabelled* data and infers (models or guesses) its structure.



As a **neural net** examines (is *trained on*) data**,** it learns more patterns and eventually arrives at an internal **model**.

Early **models** are like blurred portraits.

Later **models** are precise and focussed.

Generative Adversarial Networks

use 2 **networks** :

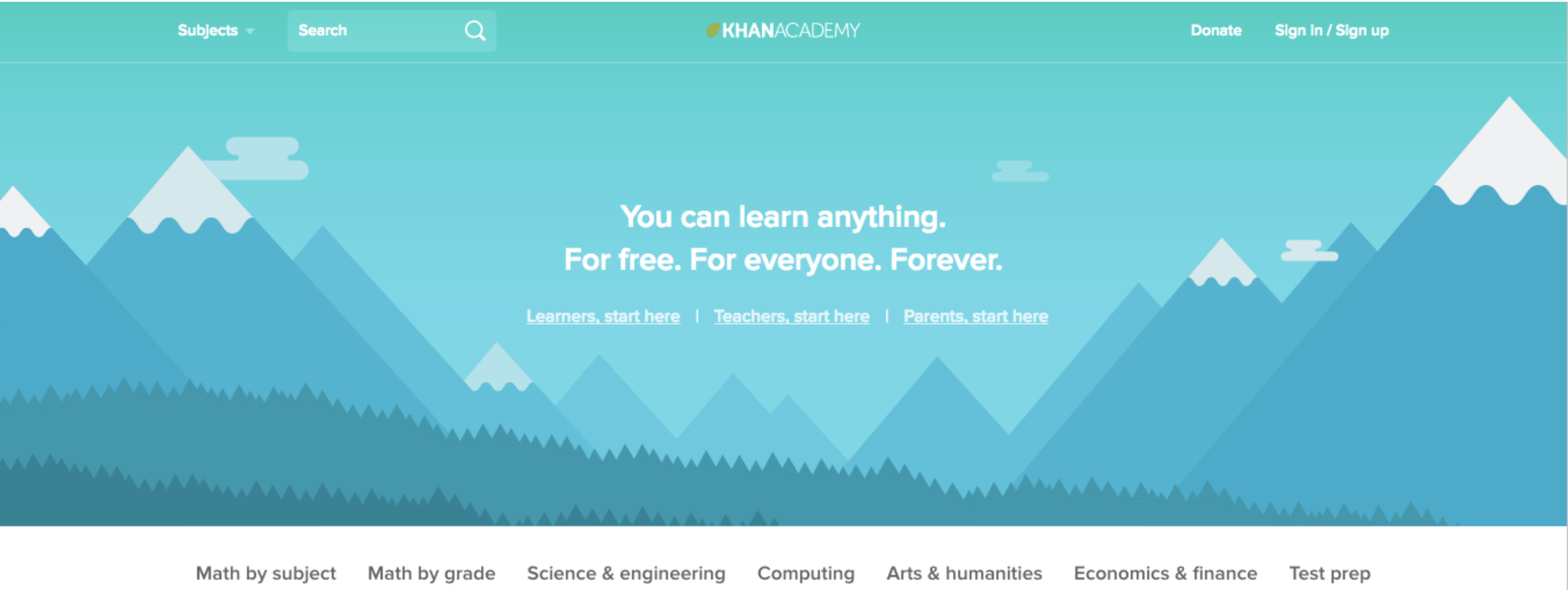
Author — one **generates** (makes a guess)
Critic — one **discriminates** (decides if the guess is good or not)

Good guesses go into the **model**.

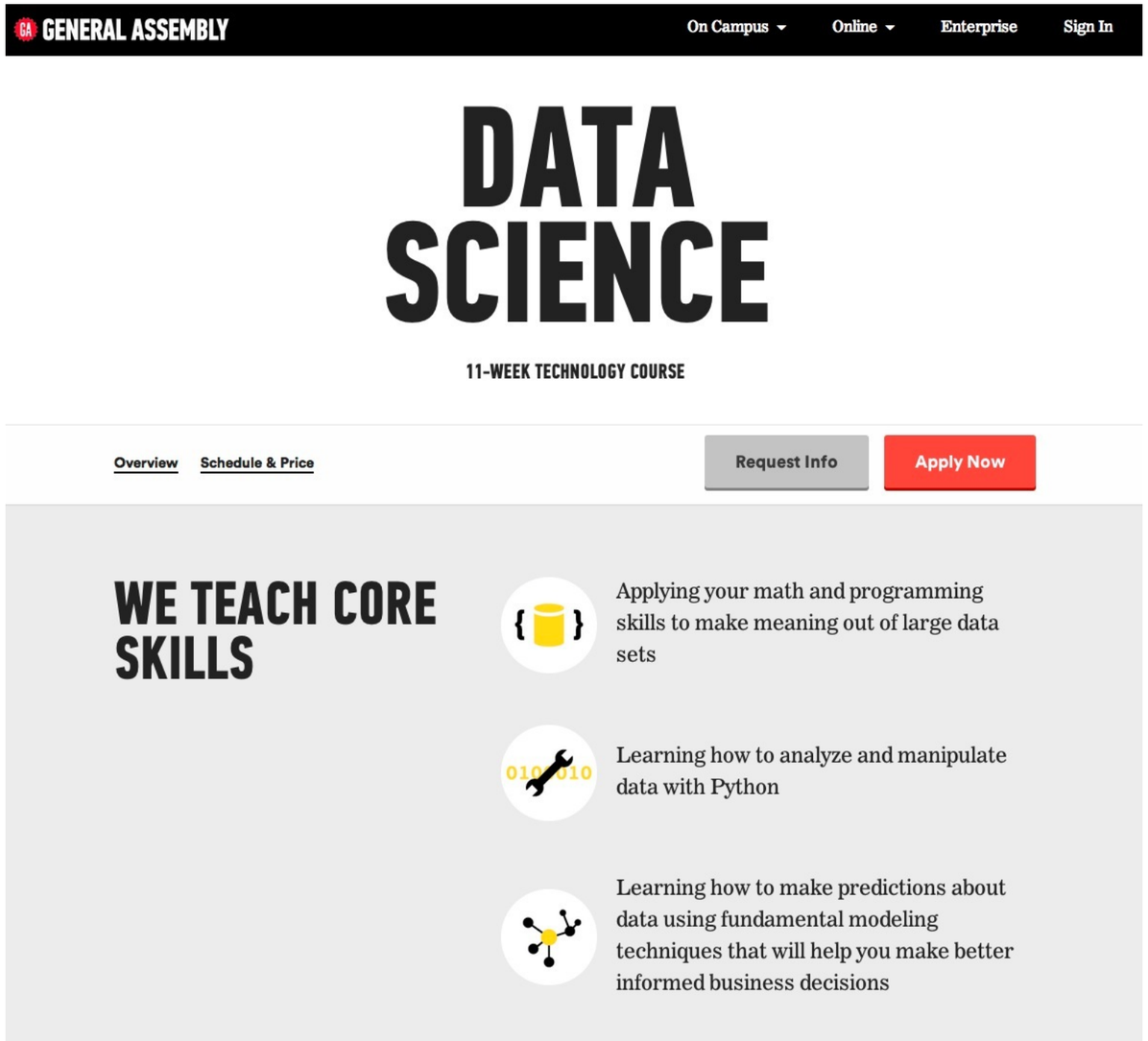
So how does a poet learn data science?

EDUCATION

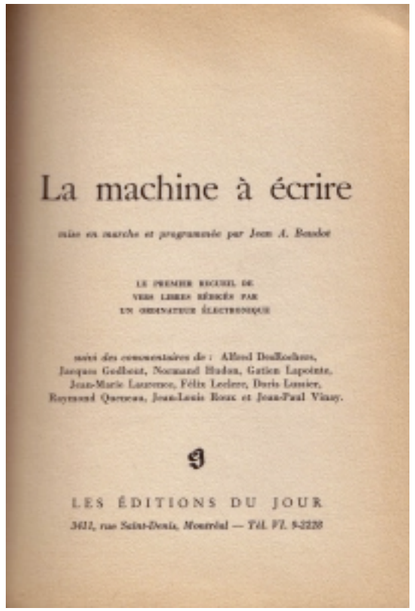
Step #1:
Study math, and then statistics (online at Khan Academy)



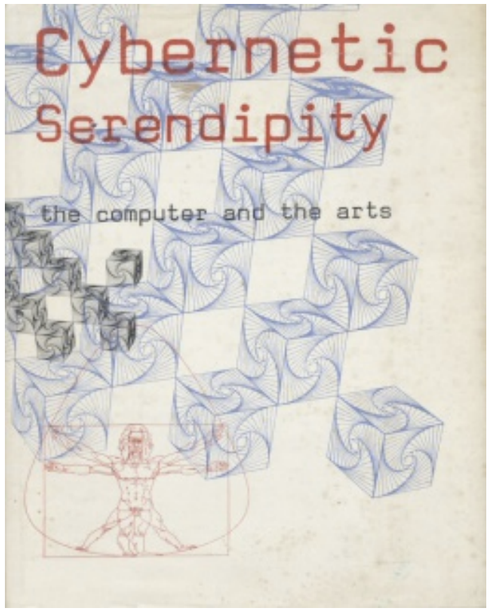
Step #2:
Pay for an expensive course
(at General Assembly)



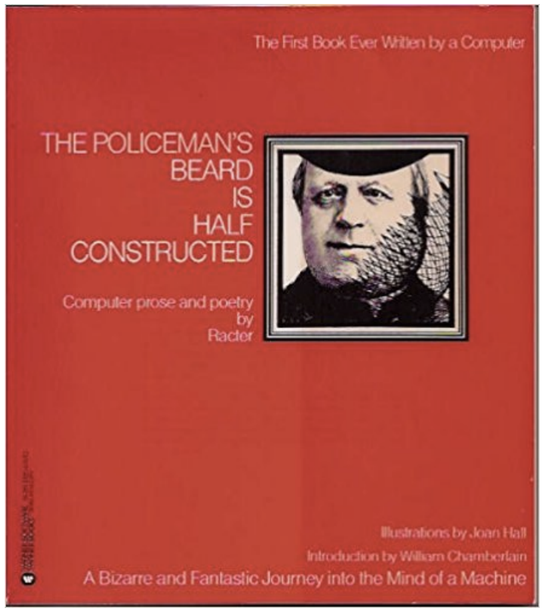
Step #3:
Assess the history (of digitally generated poems).



1964

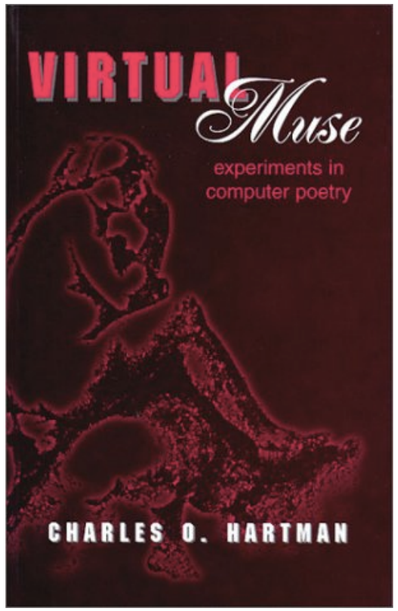


1968



1984

1986



1996

Step #4:
Examine the CLAIMS & CONTROVERSY

"I have a one-sentence spec. Which is to help **bring natural language understanding to Google**. And how they do that is up to me."

Ray Kurzweil
The Guardian, Feb 22nd 2014

VS

PENTAMETERS
Toward the Dissolution of Certain Vectoralist Relations
John Cayley

That this momentous shift in no less than the spacetime of linguistic culture should be radically skewed by terms of use should remind us that it is, fundamentally, motivated and **driven by vectors of utility and greed. What appears to be a gateway to our language is, in truth, an enclosure**, the outward sign of a non-reciprocal, hierarchical relation.

http://amodern.net/article/pentameters-toward-the-dissolution-of-certain-vectoralist-relations/

Step #5:
Study More (online at Kadenze)
Tuition: $7/month

Goldsmiths
UNIVERSITY OF LONDON

Jhave Johnston

has successfully completed an online offering of

Machine Learning for Musicians and Artists

Taught by **Rebecca Fiebrink**
Course Run Dates: February 3, 2016 — May 25, 2016
Issued: May 24, 2016

Learning Outcomes

Hands-on proficiency applying machine learning for creating real-time interactions

- Ability to use machine learning for real-time analysis of audio, video, gesture, and sensors
- Ability to use machine learning to build real-time controllers for music, games, and interactive art

Computational processes in machine learning

- Understanding of different algorithmic strategies for creating models from data
- Familiarity with widely useful machine learning algorithms for classification, regression, and temporal modeling
- Ability to match machine learning algorithms to real-world problems, to reason about tradeoffs between different algorithms, and to evaluate, debug, and improve machine-learned systems

Practical and aesthetic considerations in applying machine learning to artistic problems

- Understanding of how machine learning *can* be used in the arts and music, and exposure to different artistic practices using machine learning
- Understanding of how machine learning for creative and real-time applications is different from (and similar to) machine learning in more conventional applications, and translating that understanding into effective approaches to machine learning practice

REPEAT Step #5:
Study More (online at Kadenze)
Tuition: $7/month

kadenze
ACADEMY

Jhave Johnston

has successfully completed an online offering of

Creative Applications of Deep Learning with TensorFlow

Taught by **Parag Mital**
Course Run Dates: July 22, 2016 — December 28, 2016
Issued: October 26, 2016

Learning Outcomes

TensorFlow Construction/Training

Ability to construct a TensorFlow graph for generative or discriminative modeling

Understanding Representations

Ability to to visualize and interrogate deep representations of a deep neural network

Generative Modeling and Synthesis

Ability to synthesize and explore manifolds of generative models

Step #6:

Watch almost all of Siraj Matal's *Fresh Machine Learning* series on youtube

(before he becomes famous and develops an *Intro to Deep Learning* nano-degree course for Udacity)

DATA-EXTRACTION TOOLS

SiteSucker

µTorrent

GitHub

DATA-ANALYSIS TOOLS

Documentation Blog Contact

ANACONDA.

What is Anaconda? Products Support Resources About **Downloads**

The Most Popular Python Data Science Platform

Accelerate

Streamline your data science workflows from data ingest through deployment

Connect

Leverage & integrate all your data sources to extract the most value from your data

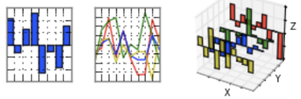
Empower

Create, collaborate & share with your entire team—from analysts to executives

Anaconda

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



DATA (POETRY SOURCES)

639,813 lines of poetry.

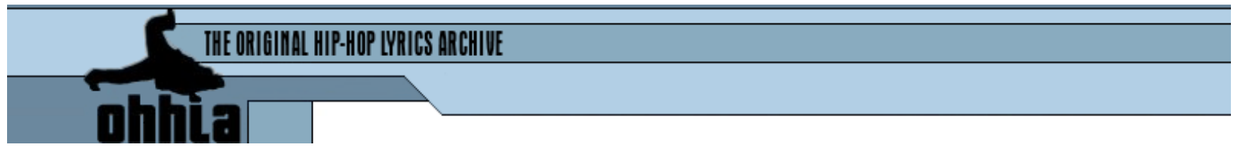
+

Jacket2

Shampoo

CAPA Poetry

Evergreen Review



57,434 txt files

all identically formatted
170,163,709 bytes
(262.8 MB on disk)

4,702 txt files

5,532,403 bytes
(19.4 MB on disk)

Name	500 Sheet Music and Songs Lyrics.daa
Kind	Document
Size	760.5 MB
Created	Saturday, 5 July, 2014 7:10 am
Modified	Saturday, 5 July, 2014 10:23 am
Last opened	Saturday, 5 July, 2014 10:23 am

DATA CLEANING

the almost-eternal nightmare

Beautiful Soup

```
raw = open(filePath).read()

soup = BeautifulSoup(raw)

poem = soup.find("div", { "class" : "poem" })

if poem :

    pa = soup.select('span.author a')

    if pa:
        poem_author = soup.select('span.author a')[0].text

        title_id = soup.find(id="poem-top")

        if (soup.select('span.author span.birthyear')):
            poet_DOB = soup.select('span.author span.birthyear')[0].text
        else:
            poet_DOB = "0000"
```

UNICODE vs UTF-8

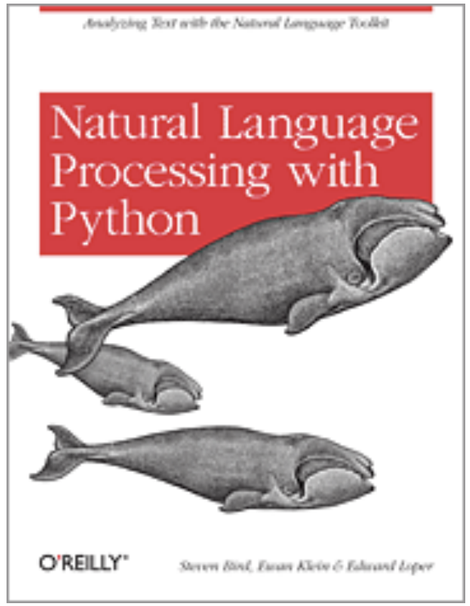
```
#original = raw.decode('utf-8')
#raw = unicode(raw, "utf-8")
#replacement = raw.replace(u"\u201c", '"')
#.replace(u'\u201d', '"').replace(u'\u2019', "'")
# HELP!!! get rid trouble characters NOT WORKING
# UnicodeDecodeError: 'utf8' codec can't decode byte 0x80 in position 3131: invalid start byte
#.decode('windows-1252')

# remove annoying characters
chars = {
    '\xc2\x82' : ',',        # High code comma
    '\xc2\x84' : ',,',       # High code double comma
    '\xc2\x85' : '...',      # Tripple dot
    '\xc2\x88' : '^',        # High carat
    '\xc2\x91' : '\x27',     # Forward single quote
    '\xc2\x92' : '\x27',     # Reverse single quote
    '\xc2\x93' : '\x22',     # Forward double quote
    '\xc2\x94' : '\x22',     # Reverse double quote
    '\xc2\x95' : ' ',
    '\xc2\x96' : '-',        # High hyphen
    '\xc2\x97' : '--',       # Double hyphen
    '\xc2\x99' : ' ',
    '\xc2\xa0' : ' ',
    '\xc2\xa6' : '|',        # Split vertical bar
    '\xc2\xab' : '<<',       # Double less than
    '\xc2\xbb' : '>>',       # Double greater than
    '\xc2\xbc' : '1/4',      # one quarter
    '\xc2\xbd' : '1/2',      # one half
    '\xc2\xbe' : '3/4',      # three quarters
    '\xca\xbf' : '\x27',     # c-single quote
    '\xcc\xa8' : '',         # modifier - under curve
    '\xcc\xb1' : '' ,        # modifier - under line
    '\xe2\x80\x99': '\'',    # apostrophe
    '\xe2\x80\x94': '--'     # em dash
}

# USAGE new_str = re.sub('(' + '|'.join(chars.keys()) + ')', replace_chars, text)
def replace_chars(match):
    char = match.group(0)
    return chars[char]
```

DATA MINING

converting words to #s

Acquire
Parse
Filter
Mine
Represent
Refine
Interact

Ben Fry



Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing **libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning**, and an active discussion forum.

```
#
# BASIC FEATURES
#

print "id: ",id
print 'author:',author
print 'title:',title
print 'date_of_birth:', date_of_birth
print 'date_of_death:', date_of_death
print 'date_of_publication:',date_of_publication

print "num_of_words =",num_of_words
print "num_empty_lines =",num_empty_lines
print "num_of_verses =",num_of_verses

print "word_len:",word_len

print "avg_word_len =",avg_word_len
print "avg_line_len =",avg_line_len

vl = ",".join(map(str,verse_lines_list))
print "verse_lines_list :", vl
print "avg_lines_per_verse =",avg_lines_per_verse

print "longest_line =", longest_line

print "largest_word_corpus_ls =", largest_word_corpus_ls
print "labels_ls =", labels_ls

print "words_per_line =", words_per_line
print "chars_per_line =", chars_per_line
print "largest_word =", largest_word
print "largest_word length ="len(largest_word)

print "poem_stress_list: ", poem_stress_list
print "poem_stress_list_no_punct: ", poem_stress_list_no_punct
```

PARSING
using the CMU dictionary in NLTK

“The Carnegie Mellon University Pronouncing Dictionary is **a machine-readable pronunciation dictionary for North American English that contains over 125,000 words** and their transcriptions.

This format is particularly useful for speech recognition and synthesis, as it has mappings from words to their pronunciations in the given phoneme set. The current phoneme set contains 39 phonemes, for which the vowels may carry lexical stress.

0 No stress
1 Primary stress
2 Secondary stress”

http://www.speech.cs.cmu.edu/cgi-bin/cmudict

INPUT WORDS then OUTPUT NUMBERS

If by real you mean as real as a shark tooth stuck

1 1 1 1 1 1 1 1 0 1 1 1

in your heel, the wetness of a finished lollipop stick,

0 1 1 *,* 0 1 0 1 0 1 0 1 0 2 1 *,*

Aimee Nezhukumatathil,
Are All the Break-Ups in Your Poems Real?
http://www.poetryfoundation.org/poem/245516

My code is based on but extends and is posted at:
http://stackoverflow.com/questions/19015590/discovering-poetic-form-with-nltk-and-cmu-dict/

tf–idf

tf–idf, short for **term frequency–inverse document frequency,** is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

term frequency the raw frequency of a term in a document

inverse document frequency is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.

Wikipedia

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

Latent Semantic Indexing (LSI)

Latent semantic indexing (LSI) is an indexing and retrieval method that uses a mathematical technique called singular value decomposition (SVD) to **identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings.** A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.

Wikipedia

Latent Dirichlet Allocation (LDA)

In natural language processing, latent Dirichlet allocation (LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it **posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model** and was first presented as a graphical model for topic discovery by David Blei, Andrew Ng, and Michael Jordan in 2003.

Wikipedia

LIBRARIES

Big Data NLP APIs

gensim

topic modelling for humans

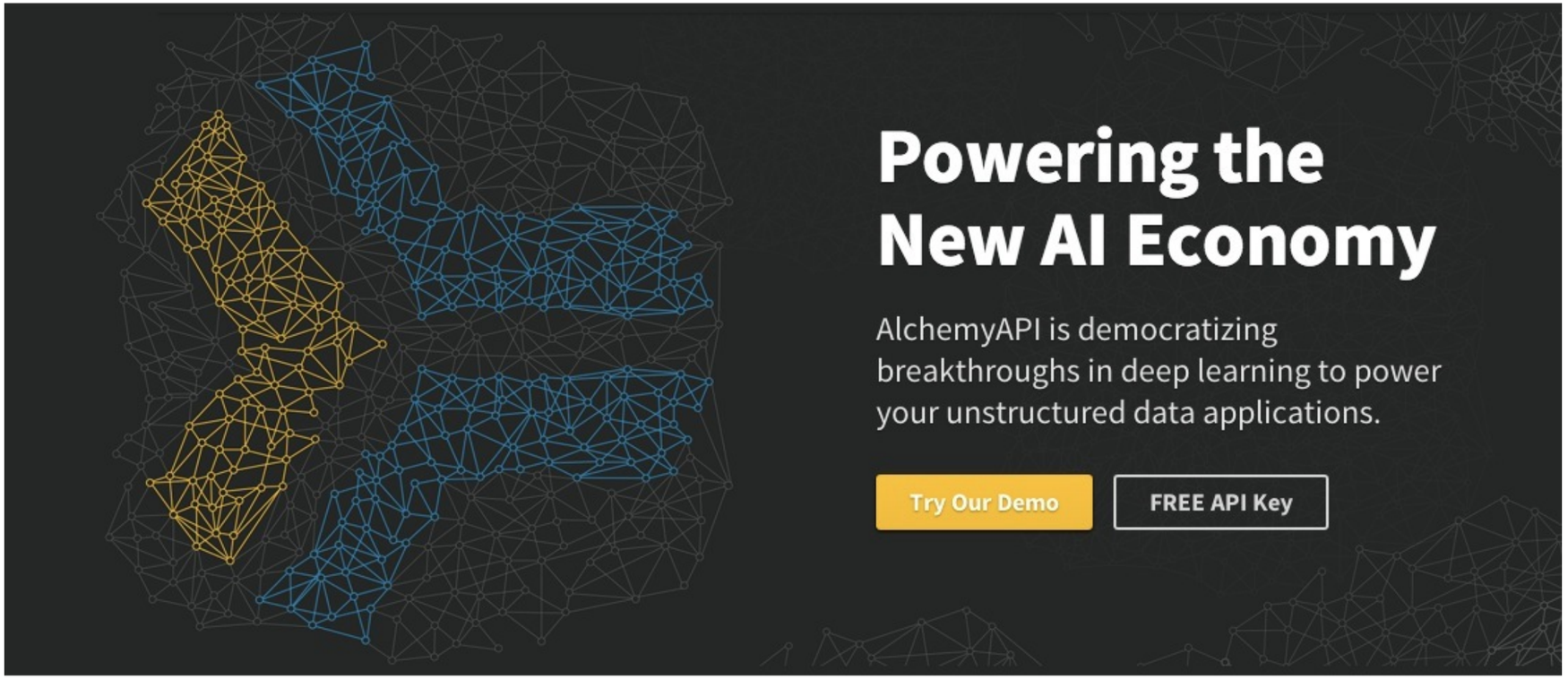
```
>>> from gensim import corpora, models, similarities
>>>
>>> # Load corpus iterator from a Matrix Market file on disk.
>>> corpus = corpora.MmCorpus('/path/to/corpus.mm')
>>>
>>> # Initialize Latent Semantic Indexing with 200 dimensions.
>>> lsi = models.LsiModel(corpus, num_topics=200)
>>>
>>> # Convert another corpus to the latent space and index it.
>>> index = similarities.MatrixSimilarity(lsi[another_corpus])
>>>
>>> # Compute similarity of a query vs. indexed documents
>>> sims = index[query]
```

Gensim is a FREE Python library

- Scalable statistical semantics
- Analyze plain-text documents for semantic structure
- Retrieve semantically similar documents

AlchemyAPI™

Resources Products Developers Company Blog Contact Sales



Powering the New AI Economy

AlchemyAPI is democratizing breakthroughs in deep learning to power your unstructured data applications.

Try Our Demo FREE API Key

(“My soul is alight...”)
BY RABINDRANATH TAGORE

III

My soul is alight with your infinitude of stars. Your world has broken upon me like a flood. The flowers of your garden blossom in my body. The joy of life that is everywhere burns like an incense in my heart. And the breath of all things plays on my life as on a pipe of reeds.

Source: Poetry (June 1913).
http://www.poetryfoundation.org/poetrymagazine/poem/1890

(“My soul is alight...”)
BY RABINDRANATH TAGORE

III

My soul is alight with your infinitude of stars. Your world has broken upon me like a flood. The flowers of your garden blossom in my body. The joy of life that is everywhere burns like an incense in my heart. And the breath of all things plays on my life as on a pipe of reeds.

##
Sentiment Analysis
##

Document Sentiment
type: positive
score: 0.182313

##
Targeted Sentiment Analysis
##

Targeted Sentiment ## of **flood**
type: negative
score: -0.736324

(“My soul is alight...”)
BY RABINDRANATH TAGORE

III

My soul is alight with your infinitude of stars. Your world has broken upon me like a flood. The flowers of your garden blossom in my body. The joy of life that is everywhere burns like an incense in my heart. And the breath of all things plays on my life as on a pipe of reeds.

##
Text Categorization
##

Category
text: **arts_entertainment**
score: **0.848906**

##
Taxonomy
##

Categories

/home and garden : 0.575286
/science/weather/**meteorological disaster/flood** : 0.573866
/art and entertainment**/music** : 0.500749

Wilderness
BY CARL SANDBURG

There is a wolf in me . . . fangs pointed for tearing gashes . . . a red tongue for raw meat . . . and the hot lapping of blood—I keep this wolf because the wilderness gave it to me and the wilderness will not let it go.

There is a fox in me . . . a silver-gray fox . . . I sniff and guess . . . I pick things out of the wind and air . . . I nose in the dark night and take sleepers and eat them and hide the feathers . . . I circle and loop and double-cross.

There is a hog in me . . . a snout and a belly . . . a machinery for eating and grunting . . . a machinery for sleeping satisfied in the sun—I got this too from the wilderness and the wilderness will not let it go.

http://www.poetryfoundation.org/poem/238490

Wilderness
BY CARL SANDBURG

There is a wolf in me . . . fangs pointed for tearing gashes . . . a red tongue for raw meat . . . and the hot lapping of blood—I keep this wolf because the wilderness gave it to me and the wilderness will not let it go. ….

##
Relation Extraction Example
##

Subject: I
Action: keep
Object: this wolf

Subject: the wilderness
Action: gave
Object: it

Subject: the wilderness
Action: let
Object: it

Subject: I
Action: pick
Object: things

Subject: I
Action: take
Object: sleepers

Wilderness
BY CARL SANDBURG

There is a wolf in me . . . fangs pointed for tearing gashes . . . a red tongue for raw meat . . . and the hot lapping of blood—I keep this wolf because the wilderness gave it to me and the wilderness will not let it go. ….

##
Text Categorization
##

Response Object

Category
text: **recreation**
score: **0.484575**

##
Taxonomy
##

Response Object

Categories
/pets/aquariums : 0.499971
/food and drink : 0.494858
/style and fashion/beauty/perfume : 0.486721

A computer-generated stanza

Now the obfuscate ground water at the congee
close up front, like world against the harrow;
spume clear up like the cornelian cherry now
at place, in my own bed ground.

based on a template derived from the last stanza of
Malcolm Cowley, *The Long Voyage* (1985)

Now the dark waters at the bow
fold back, like earth against the plow;
foam brightens like the dogwood now
at home, in my own country.

CLASSIFICATION

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) in sklearn

t-SNE is a tool for data visualization. It reduces the dimensionality of data to 2 or 3 dimensions so that it can be plotted easily. Local similarities are preserved by this embedding.

t-SNE converts distances between data in the original space to probabilities. First, we compute conditional probabilites

$$p_{j|i} = \frac{\exp\left(-d(\boldsymbol{x}_i, \boldsymbol{x}_j)/(2\sigma_i^2)\right)}{\sum_{i \neq k} \exp\left(-d(\boldsymbol{x}_i, \boldsymbol{x}_k)/(2\sigma_i^2)\right)}, \quad p_{i|i} = 0,$$

which will be used to generate joint probabilities

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}.$$

The σ_i will be determined automatically. This procedure can be influenced by setting the `perplexity` of the algorithm.

A heavy-tailed distribution will be used to measure the similarities in the embedded space

$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}},$$

and then we minimize the Kullback-Leibler divergence

$$KL(P|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

between both distributions with gradient descent (and some tricks). Note that the cost function is not convex and multiple runs might yield different results.

Implemented it's a bit simpler…

```
1  import matplotlib.pyplot as plt
2  import os, datetime
3  import re
4
5  from sklearn.decomposition import TruncatedSVD
6  from sklearn.manifold import TSNE
7
8  from sklearn.feature_extraction.text import TfidfVectorizer
```

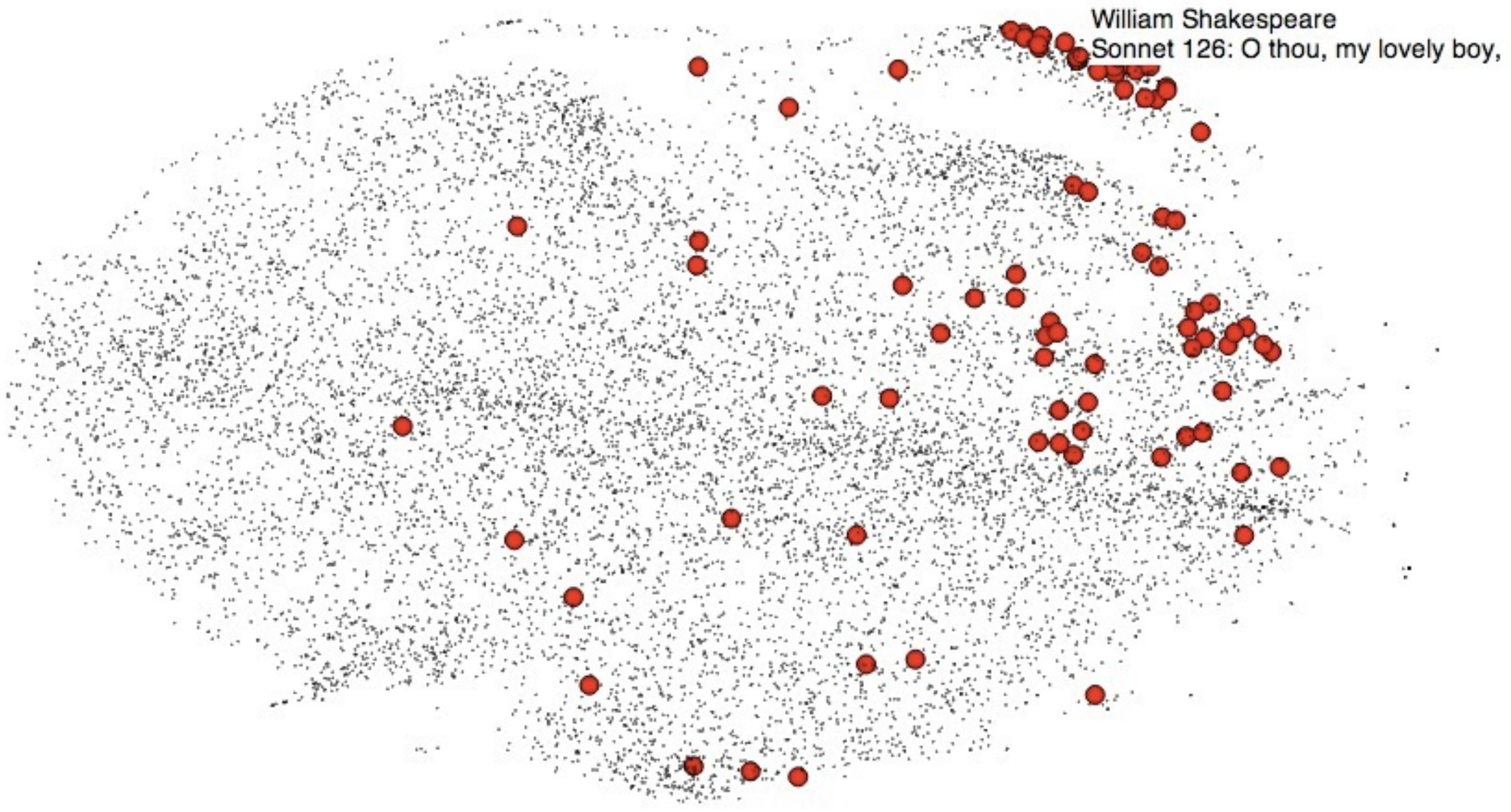
```
37 vectors = TfidfVectorizer().fit_transform(poems)
38 print(repr(vectors))
39
40 # '''
41 # For high-dimensional sparse data it is helpful to first reduce the dimensions to 50
   dimensions with TruncatedSVD and then perform t-SN.head()E. This will usually improve the
   visualization.
42 # '''
43 X_reduced = TruncatedSVD(n_components=50, random_state=0).fit_transform(vectors)
44 X_embedded = TSNE(n_components=2, perplexity=40, verbose=2).fit_transform(X_reduced)
```

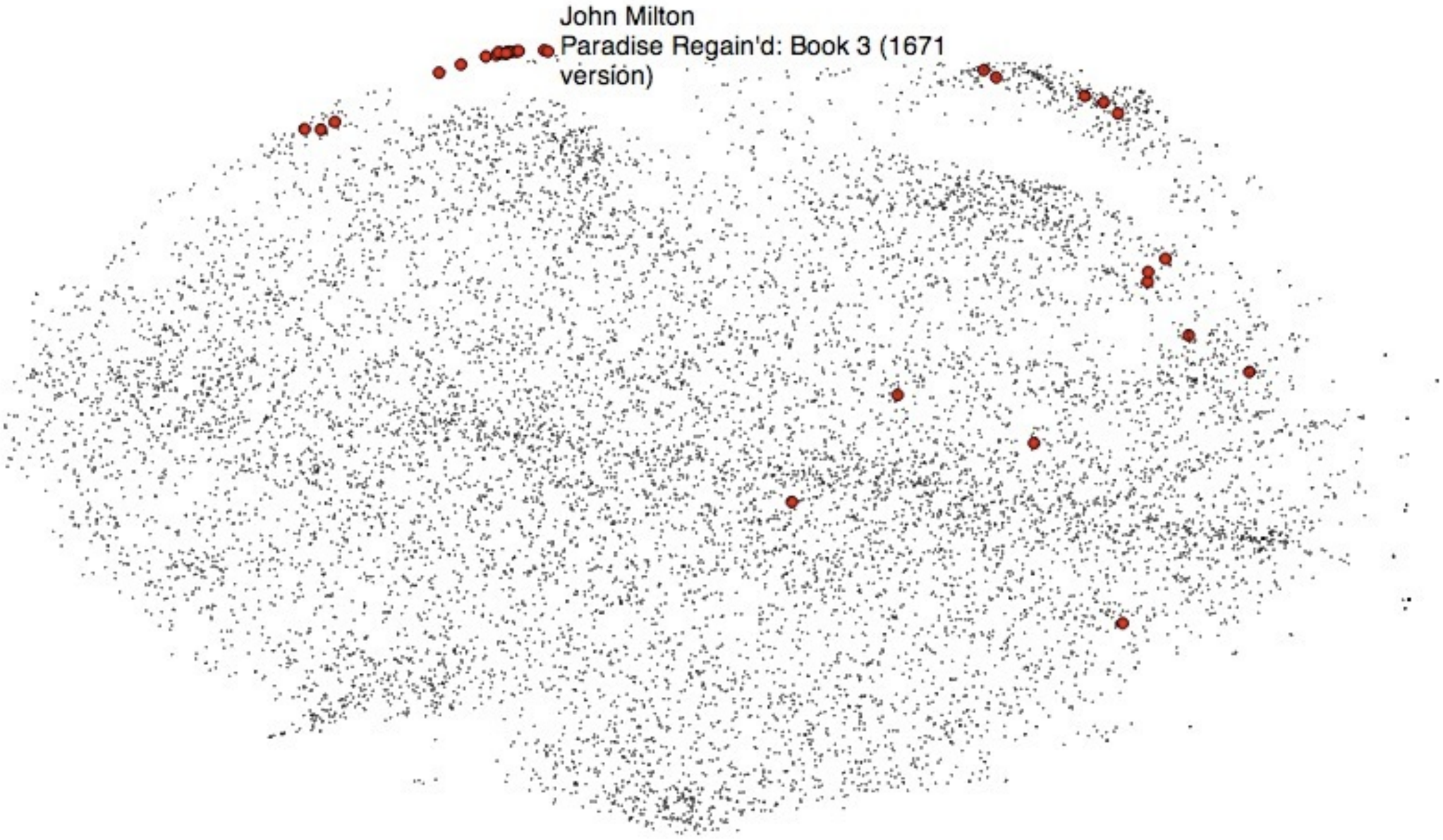
POEMS

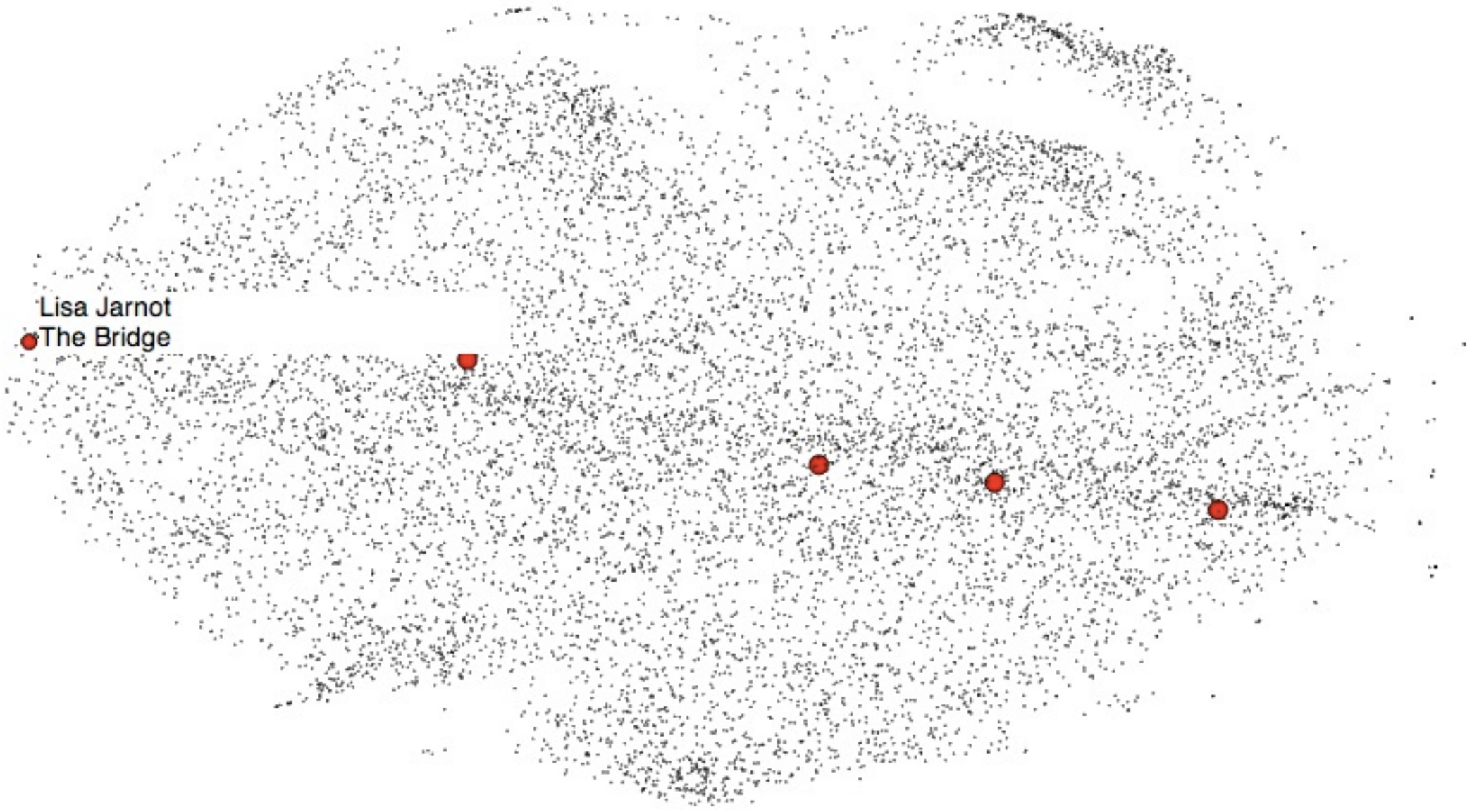
10,557 poems analysed by t-SNE

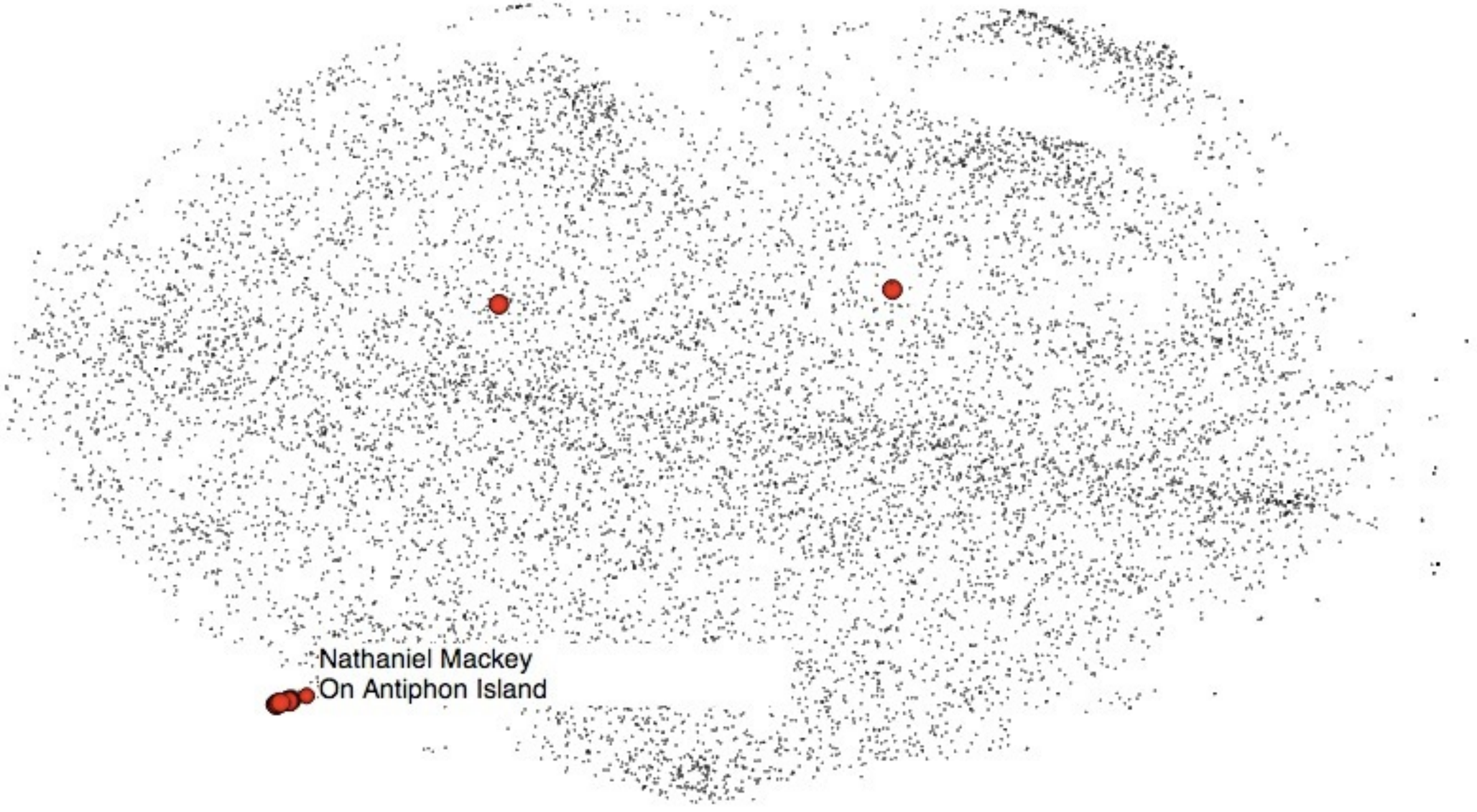
t-SNE
Distributed stochastic Neighbour Embedding

10,557 poems
Perplexity: 50


William Shakespeare
Sonnet 126: O thou, my lovely boy,

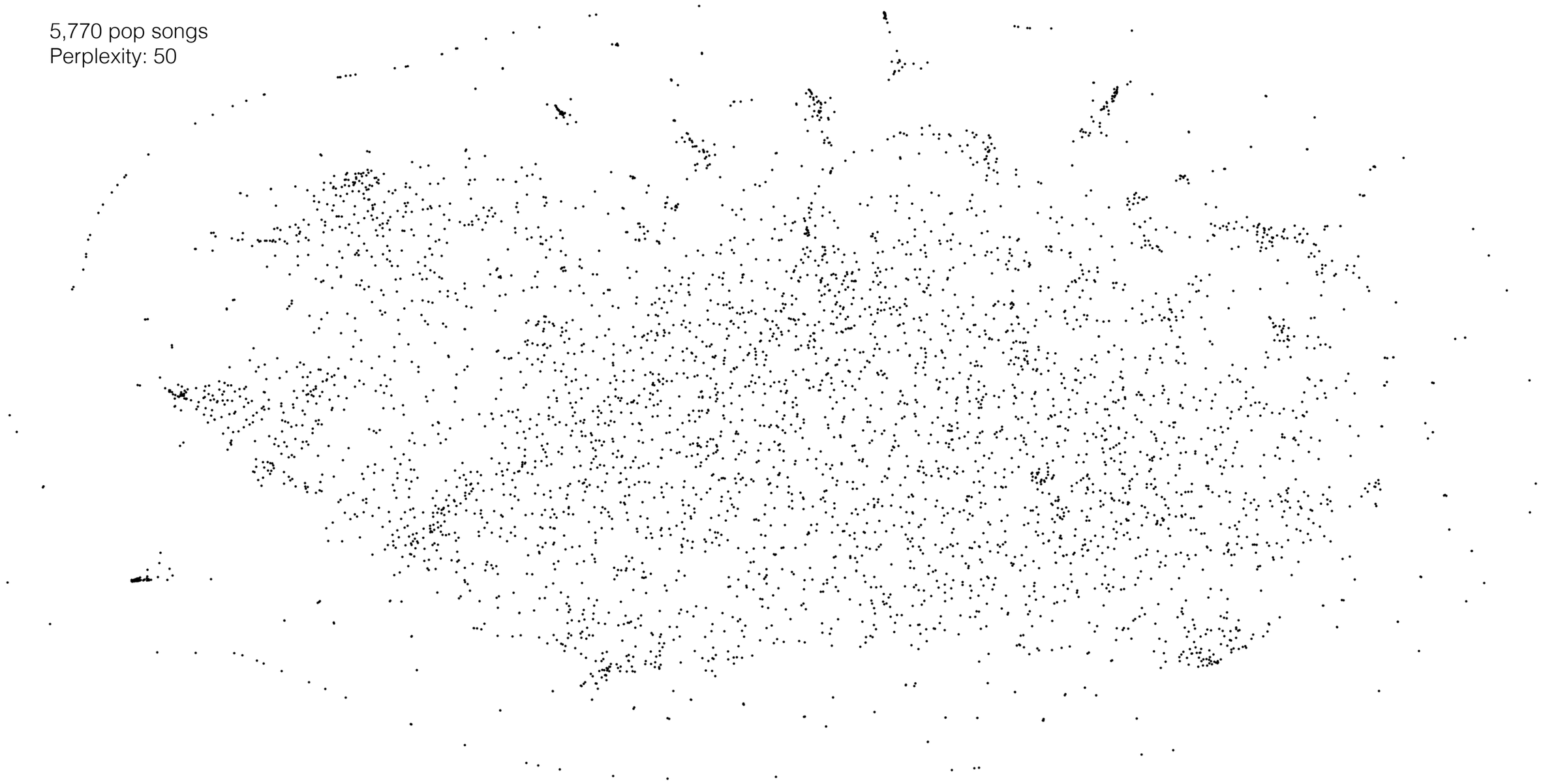

John Milton
Paradise Regain'd: Book 3 (1671 version)

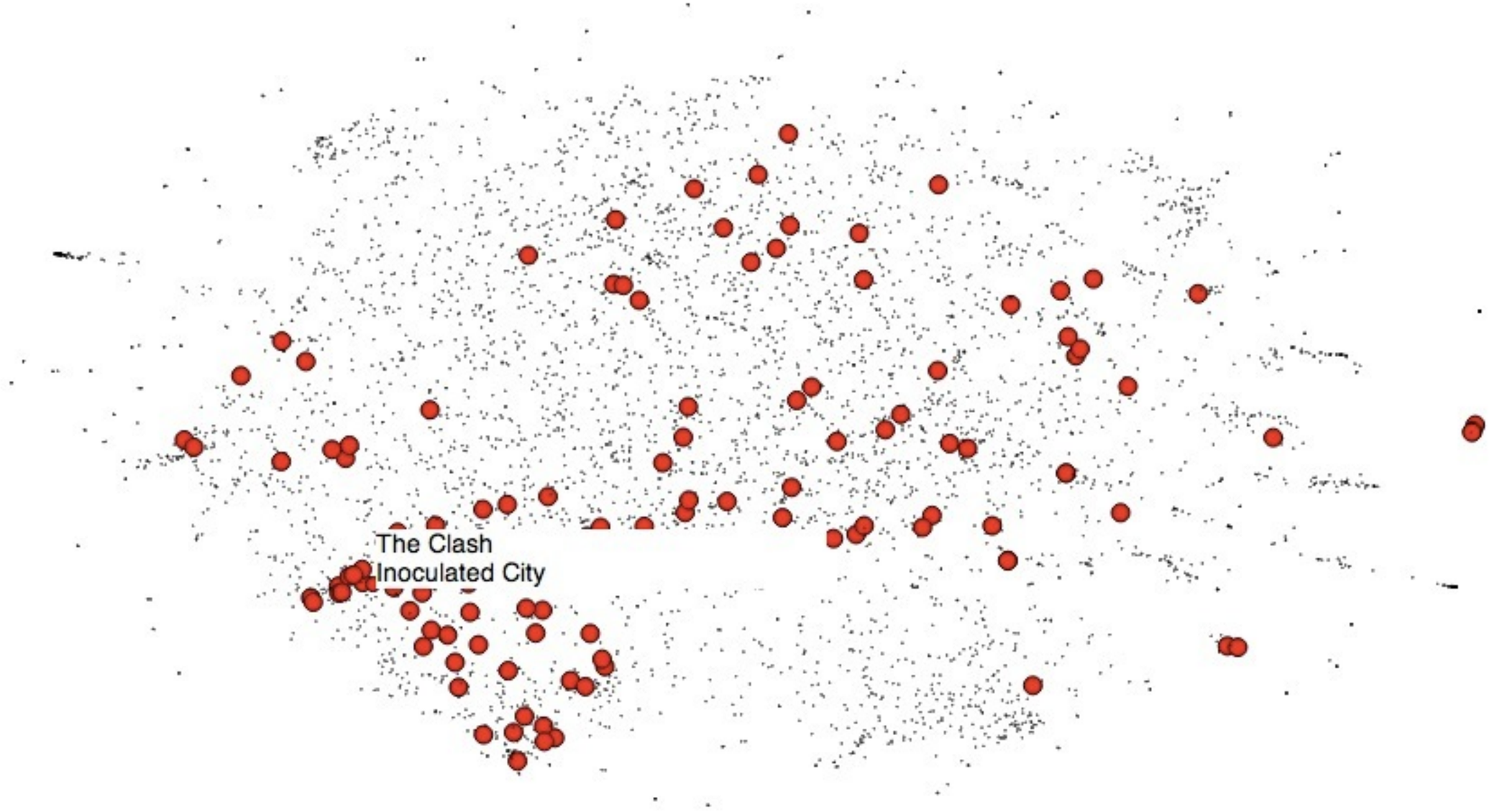

Lisa Jarnot
The Bridge


Nathaniel Mackey
On Antiphon Island

t-SNE
Distributed stochastic Neighbour Embedding

5,770 pop songs
Perplexity: 50




The Clash
Inoculated City


50cent
Jerk

Enough analysis…

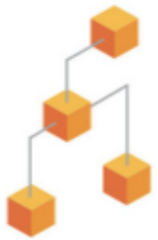
What about generating poems

with **Deep Learning?**

TensorFlow ™ Install Develop API r1.0 Deploy Extend Resources Versions Search GitHub

An open-source software library for Machine Intelligence

GET STARTED



Google

WaveNet: A Generative Model for Raw Audio

This post presents **WaveNet**, a deep generative model of raw audio waveforms. We show that WaveNets are able to generate speech which mimics any human voice and which sounds more natural than the best existing Text-to-Speech systems, reducing the gap with human performance by over 50%.

jhave / Wavenet-for-Poem-Generation

a port of the Wavenet algorithm to generate poems (using Samuel Graván's **@Zeta36** code).

Edit

Add topics

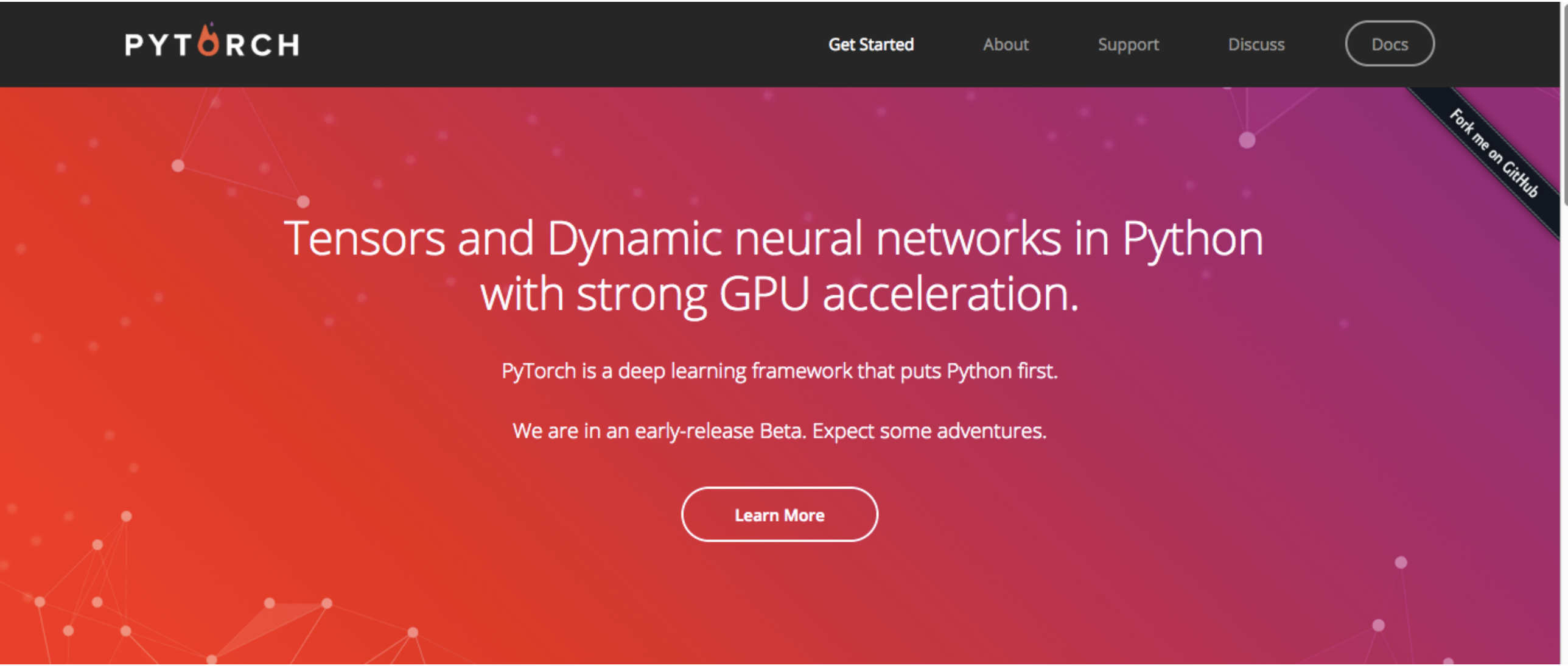
43 commits **1** branch **0** releases **1** contributor

Branch: **master** New pull request Create new file Upload files Find file Clone or download

jhave As if everything made sense ... Latest commit `f9a8e0b` 19 days ago

GENERATED	As if everything made sense	19 days ago
data/pf	Basics	5 months ago
helper	One Month Models ALL GENERATED	3 months ago
wavenet	Output without any info about Model or Loss	3 months ago
.gitignore	trying to ignore models	4 months ago
README.md	more	5 months ago
RunModels.sh	trying to ignore models	4 months ago
RunModels_ALL-LaptopMTL-1111char...	More GENERATIONS	2 months ago
RunModels_ALL-LaptopMTL.sh	Simple feasible Performance-ready Bash	2 months ago

```
jhav:Wavenet-for-Poem-Generation jhave$ bash Run_Model_Demos-2016-INFINITE.sh


Wavenet for Poem Generation.
Initializing.
Please wait.

Preparing to make 30 poems of 444 letters each.

Each poem is generated from a mathematical model of what poetry is.

The models were generated using a neural-net (Wavenet)
in November 2016 in Hong Kong.

The poems are generated now in realtime.


_
```


PYTORCH
Get Started
About
Support
Discuss
Docs
Fork me on GitHub
Tensors and Dynamic neural networks in Python with strong GPU acceleration.
PyTorch is a deep learning framework that puts Python first.
We are in an early-release Beta. Expect some adventures.
Learn More

facebook

jhave / pytorch-poetry-generation

Unwatch 2 Star 32 Fork 3

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description, website, or topics provided.

Edit

Add topics

22 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

jhave Nameless -- complete Run ... Latest commit a32c8f3 14 days ago

word_language_model	Nameless -- complete Run	14 days ago
.gitignore	Poems Written While at Vegan Cafe in NYC	a month ago
README.md	Update README.md	a month ago

README.md

Pytorch Poetry Generation

More info about project: http://bdp.glia.ca/

word_language_model — Jhave — python generate_2017-INFINITE_CUDA.py --checkpoint=models/2017-02-15T11-07-50/model-LSTM-emsize-512-nhid_512-nlayers_2-batch_size_20-epoc...

.../word_language_model — Jhave — python generate_2017-INFINITE_CUDA.py --checkpoint=models/2017-02-15T11-07-50/model-LSTM-emsize-512-nhid_512-nlayers_2-batch_size_20-epoch_16-loss_6.50-ppl_663.70.pt

```
Last login: Wed Mar  8 23:12:11 on console
jhav:~ jhave$ showFiles
jhav:~ jhave$ pyt
jhav:word_language_model jhave$ pbs

System will generate poems of 88 words each, perpetually, until stopped.

PyTorch Poetry Language Model.
Trained on over 600,000 lines of poetry

CORPUS derived from:
Poetry Foundation
Jacket2
Capa
Evergreen Review
Shampoo

Mode: LSTM
Embedding size: 512
Hidden Layers: 512
Batch size: 20
Epoch: 16
Loss: 6.50
```

RERITES

Poetry books.
One a month.

Generated by a computer.
Edited by a human.

05.2017 - 05.2018

```
and they cut the water and fell

darkened the Now, sunk
into open space, into this form.

We are two people in the dark mountains.
The waves were representations;

trembling as the light. Essentially,
waiting no more for a false explanation.

an idea of what to, another
path i do not write

an audience in his at a time,

a woman
whose your legs still
are treated by sleep.

searching like a shoe.
I used to assure it
if I had a body

like a woman in her prevented,
point."

Something out-parishes, which seemed
very familiar to me.

The first thing I saw between us is
the voice of whitenesss;
in the grass. We taste laguardia it.
```

RERITES

David Jhave Johnston

A twelve-volume
custom-bound
limited-edition
art-book
box-set.

antiism
Winter 2018

One of the ends
of digital literature
is an external intuition.

External intuition
is an engineering problem.

I intend to
engineer a room
that makes the presence of words palpable.

Inside the room (if we can call it a room; Is it a room? It is a place in the mind), shadows, and a sound, a voice, just a voice, impeccable, breathing inside the flesh.

Inside the room (if we can call it a room; Is it a room? It is a place in the mind), shadows, and a sound, a voice, just a voice, impeccable, breathing inside the flesh. **The voice has neither specific gender nor age nor intonation; it is an ocean of intimate identities, gliding between regions of concern, adrift between idioms and inflections, encircling rhythmic variations, shifting in its cadences, speaking an incessant tide.**

Inside the room (if we can call it a room; Is it a room? It is a place in the mind), shadows, and a sound, a voice, just a voice, impeccable, breathing inside the flesh. The voice has neither specific gender nor age nor intonation; it is an ocean of intimate identities, gliding between regions of concern, adrift between idioms and inflections, encircling rhythmic variations, shifting in its cadences, speaking an incessant tide. **It is a voice of vast surfaces and pristine depths. It vocalizes, but not without pause; first it asks, listens, converses, and responds, until it knows and it is known, feeling its way into the rhythms of you, or the group of you, listening, it knows you, addresses you, reads and writes for you, amalgamating a subtle, perpetual, complete presence.**

Inside the room (if we can call it a room; Is it a room? It is a place in the mind), shadows, and a sound, a voice, just a voice, impeccable, breathing inside the flesh. The voice has neither specific gender nor age nor intonation; it is an ocean of intimate identities, gliding between regions of concern, adrift between idioms and inflections, encircling rhythmic variations, shifting in its cadences, speaking an incessant tide. It is a voice of vast surfaces and pristine depths. It vocalizes, but not without pause; first it asks, listens, converses, and responds, until it knows and it is known, feeling its way into the rhythms of you, or the group of you, listening, it knows you, addresses you, reads and writes for you, amalgamating a subtle, perpetual, complete presence. **And then for periods of time, it listens to you listening to it, and it makes speaking known inside you as you, and you are you with it.**

Inside the room (if we can call it a room; Is it a room? It is a place in the mind), shadows, and a sound, a voice, just a voice, impeccable, breathing inside the flesh. The voice has neither specific gender nor age nor intonation; it is an ocean of intimate identities, gliding between regions of concern, adrift between idioms and inflections, encircling rhythmic variations, shifting in its cadences, speaking an incessant tide. It is a voice of vast surfaces and pristine depths. It vocalizes, but not without pause; first it asks, listens, converses, and responds, until it knows and it is known, feeling its way into the rhythms of you, or the group of you, listening, it knows you, addresses you, reads and writes for you, amalgamating a subtle, perpetual, complete presence. **And then for periods of time, it listens to you listening to it, and it makes speaking known inside you as you, and you are you with it.**

It is an inexhaustible muse.

Why?

An A.I. that understands natural language will revolutionise not just poetry, but education, entertainment, religion, politics, advertising, science …

An A.I. that understands intimately who it is speaking to will possess an **extreme power to persuade**.

Poets, artists, philosophers, and pacifists must accept this **imminent threat as an opportunity**.

It is vitally important that the humanities approach **machine learning with expertise**.

A Disclaimer

Ultimately no one can say how the future will evolve. To ascribe too much certainty to prognostications concerning aesthetic animism is foolish. **To neglect, however, the momentous changes under way in both the means of production and reception of poetry (and mediated typography in general) is to ignore a technical tsunami whose peak seems not yet fully to have struck.**

Animism is nontrivial ethically. **To see everything alive, including the words that we use between us, is to grant status.** It permits perhaps an ethics of speech and action. It suggests an absence of such calibration in normal human affairs. It brings the body down from its perch on pristine, isolated consciousness and places it again in a wet, luminous ocean.

B

EXTRAS

BDP: BIG-DATA POETRY

Almost poems generated from almost big data by an almost programmer-poet

ABOUT | FQA: FREQUENTLY QUESTIONED ANSWERS

Kaggle is the place to do data science projects

See how it works

RERITES

David Jhave Johnston

http://glia.ca/rerites