Text Ascent

How It Works

Sherry Yang

About Text Ascent

Text Ascent uses unsupervised learning to help you discover more about a topic you are reading about. Traverse from base camp to summit and discover content based on text complexity

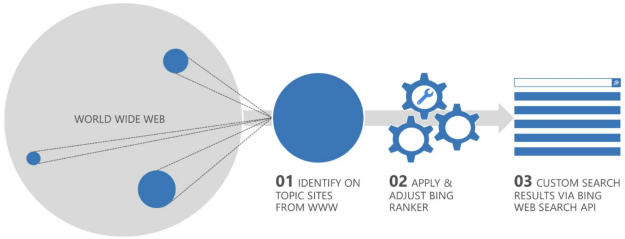
Share what you're learning with friends.

Let's **traverse**!

In the world of the one-shot-search

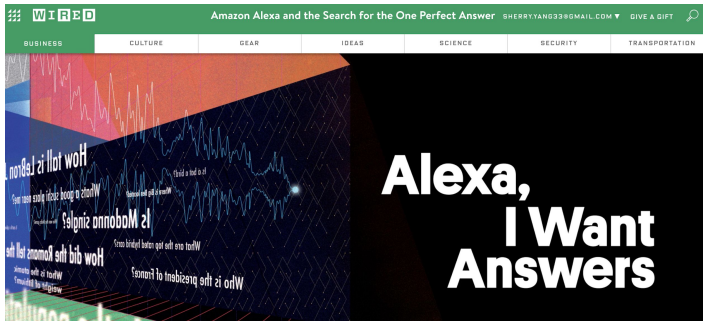
I find that, despite all of the information being thrown at us all constantly, it can be difficult to find solid resources for developing baseline knowledge. How do you find ways to boost your knowledge, whether it be the specific jargon of your industry or just something you're curious about? #resources #boost #learn

Search

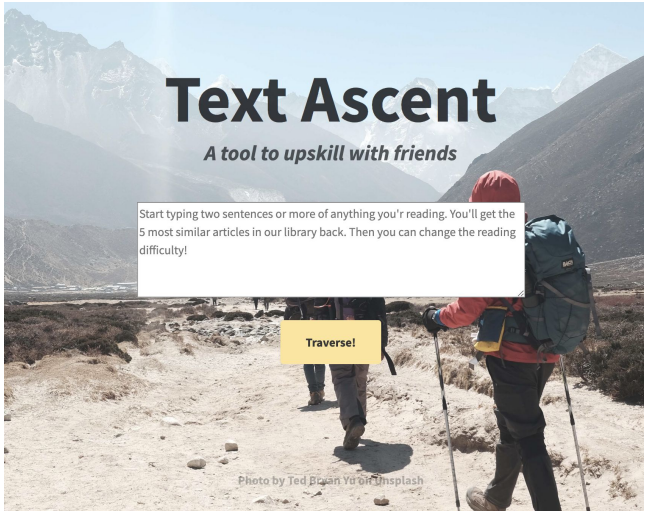


Source: "Build Your Own Web Search", Bing Blogs

Voice



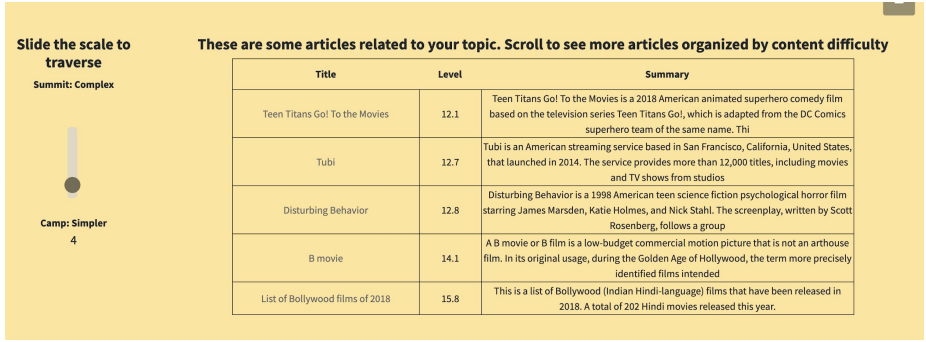
Source: Amazon Alexa and the Search for one perfect answer, *Wired*



How it works:

Takes in text and finds the most similar text in the corpus.

Then gives you the option to traverse upon a reading gradient.



Slide the scale to traverse

Summit: Complex

Camp: Simpler

4

These are some articles related to your topic. Scroll to see more articles organized by content difficulty

Title	Level	Summary
Teen Titans Go! To the Movies	12.1	Teen Titans Go! To the Movies is a 2018 American animated superhero comedy film based on the television series Teen Titans Go!, which is adapted from the DC Comics superhero team of the same name. Thi
Tubi	12.7	Tubi is an American streaming service based in San Francisco, California, United States, that launched in 2014. The service provides more than 12,000 titles, including movies and TV shows from studios
Disturbing Behavior	12.8	Disturbing Behavior is a 1998 American teen science fiction psychological horror film starring James Marsden, Katie Holmes, and Nick Stahl. The screenplay, written by Scott Rosenberg, follows a group
B movie	14.1	A B movie or B film is a low-budget commercial motion picture that is not an arthouse film. In its original usage, during the Golden Age of Hollywood, the term more precisely identified films intended
List of Bollywood films of 2018	15.8	This is a list of Bollywood (Indian Hindi-language) films that have been released in 2018. A total of 202 Hindi movies released this year.

Use Case

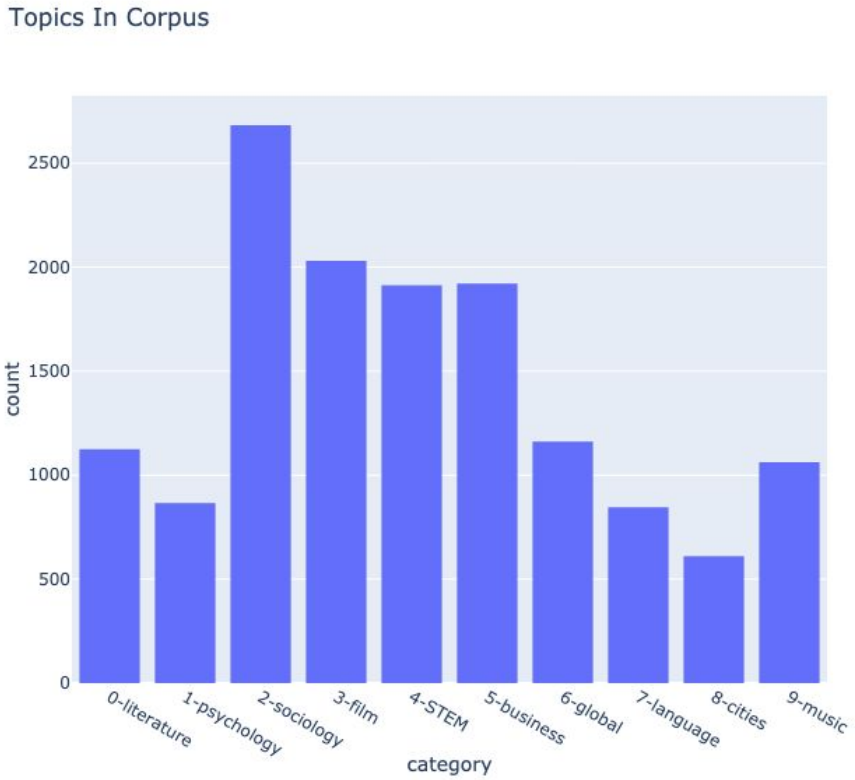
Caley comes across an article on data science that sounds interesting, but contains a lot of themes that she has not learned about before. Using Text Ascent, Caley can input the text and search along that topic for more or less complex content.

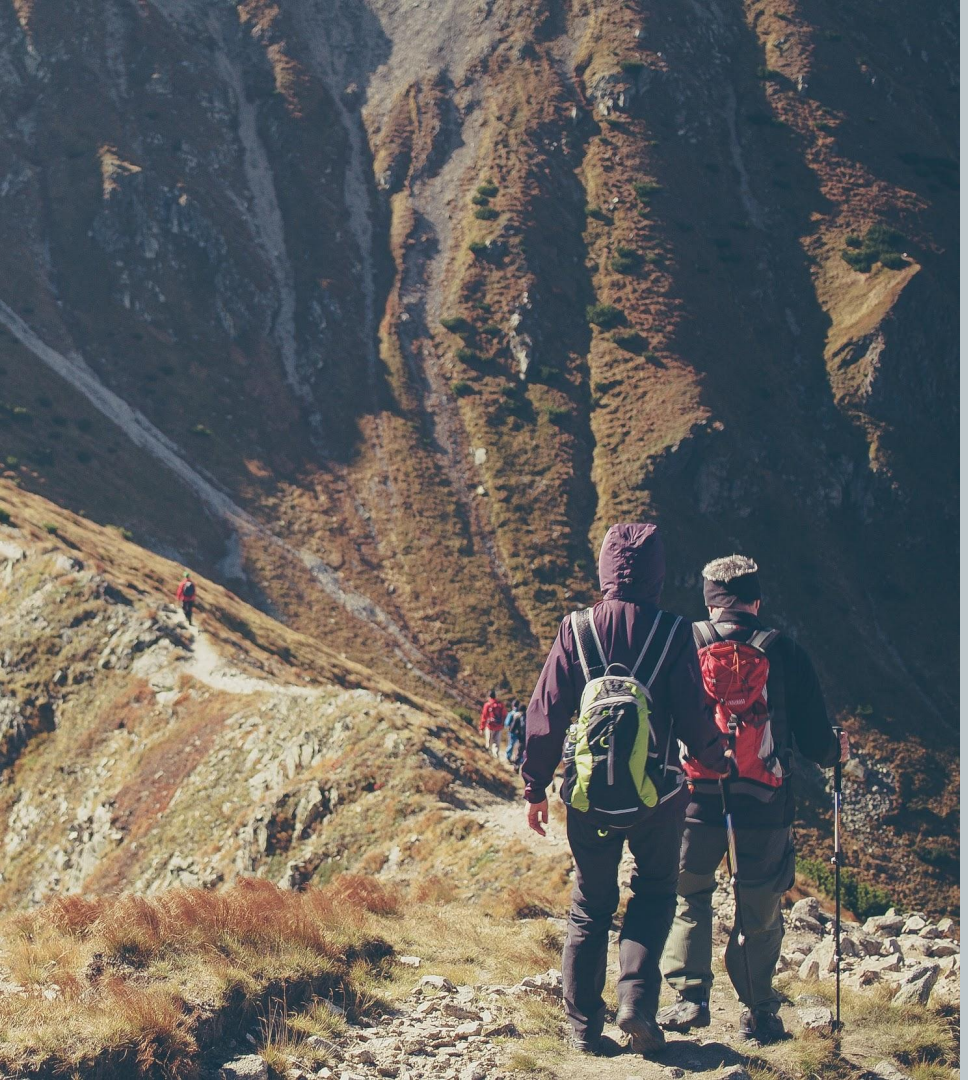
Data Collection & Understanding

Topics in corpus

Based on LDA topic model of 14,000+ wikipedia articles from the wikipedia API

10 topics created by using model to label top topic for each article





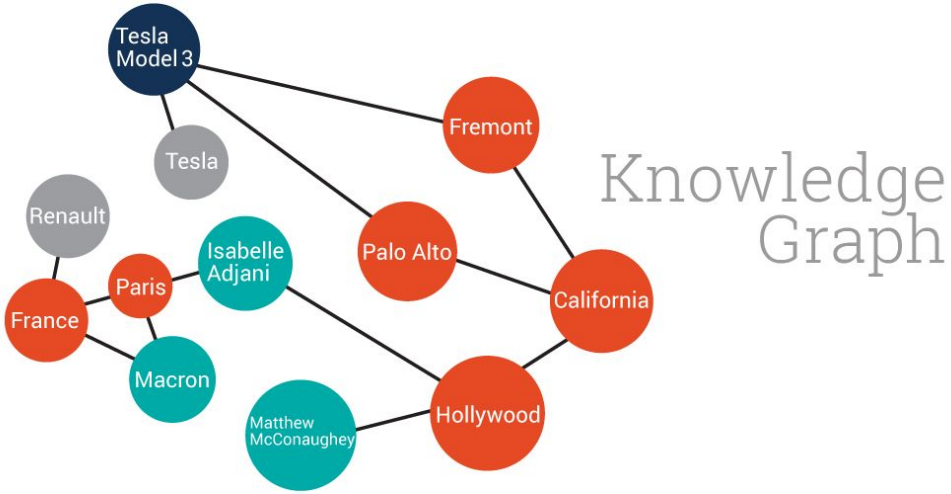
Latent Dirichlet Allocation Modeling

How It Works

Why it didn’t work for the model

Model Creation

Knowledge Graph Representation



This is the concept I started with when thinking about my model.

Source: What is a knowledge graph?, Ontotext

TFIDF Vectorization

TFIDF Vectorization from sklearn

1. TF - term frequency
2. IDF - inverse document frequency

It is calculated by taking the **log** of {number of docs in your corpus divided by the number of docs in which this term appears}.
So for "*the*", the ratio will be close to 1 since it is present in most (all) docs. Therefore, the log will take this ratio to 0.

3. Weights are given to the frequency

Source: How do TFIDF Vectors Work In Layman's Terms?, Quora

Distance to most similar text

Top 20 features

Cosine distance to most similar text

```
feature_ranking[:20]
```

```
array([ 525, 1632,  446, 1759, 1162, 1694, 1635,  118, 1043,  860, 1455,
       1442, 1129, 1331,  228, 1916,  127, 1483,  674, 1786])
```

```
vocab_arr = get_vocab_arr(vec)
vocab_arr[feature_ranking[:20]]
```

```
array(['data', 'science', 'computer', 'statistics', 'mathematics',
       'skills', 'scientists', 'able', 'knowledge', 'good', 'programming',
       'processing', 'machine', 'pattern', 'aspect', 'useful', 'achieve',
       'purposes', 'engineering', 'subjects'], dtype=object)
```

```
distances = cdist(
    get_top_k_vector(sample_vector, feature_ranking),
    get_top_k_vector(corpus_vectors, feature_ranking),
)
```

```
get_top_k_vector(sample_vector, feature_ranking)
```

```
array([[0.83470286, 0.22166612, 0.18837988, 0.14557046, 0.13767567,
        0.13719084, 0.13478882, 0.1069378 , 0.10534487, 0.09932174,
        0.07654316, 0.07365873, 0.07297837, 0.07146458, 0.06887271,
        0.06806918, 0.06750797, 0.06658652, 0.06529604, 0.06506513]])
```

Grading text:

Using TextStat's module

Flesch-Kincaid Grade of the given text. This is a grade formula in that a score of 9.3 means that a ninth grader would be able to read the document.

Deployment

How to deploy Text Ascent

Current deployment is with an AWS EC2 instance with flask integration with data stored in AWS S3

Go to traverse.sherzyang.com

Add your own text by cloning my repo

Future Uses

Demo

Invite corpus input

Amazon alexa skill where you can get summaries of reading by gradient

Sherry Yang

Thanks!

Data Scientist, Developer Advocate

Use the app: traverse.sherzyang.com

Reproduce the product:
github/sherzyang/text-ascent

Let's grab tea & discuss data science, lifelong learning and the outdoors.

sherry.yang33@gmail.com