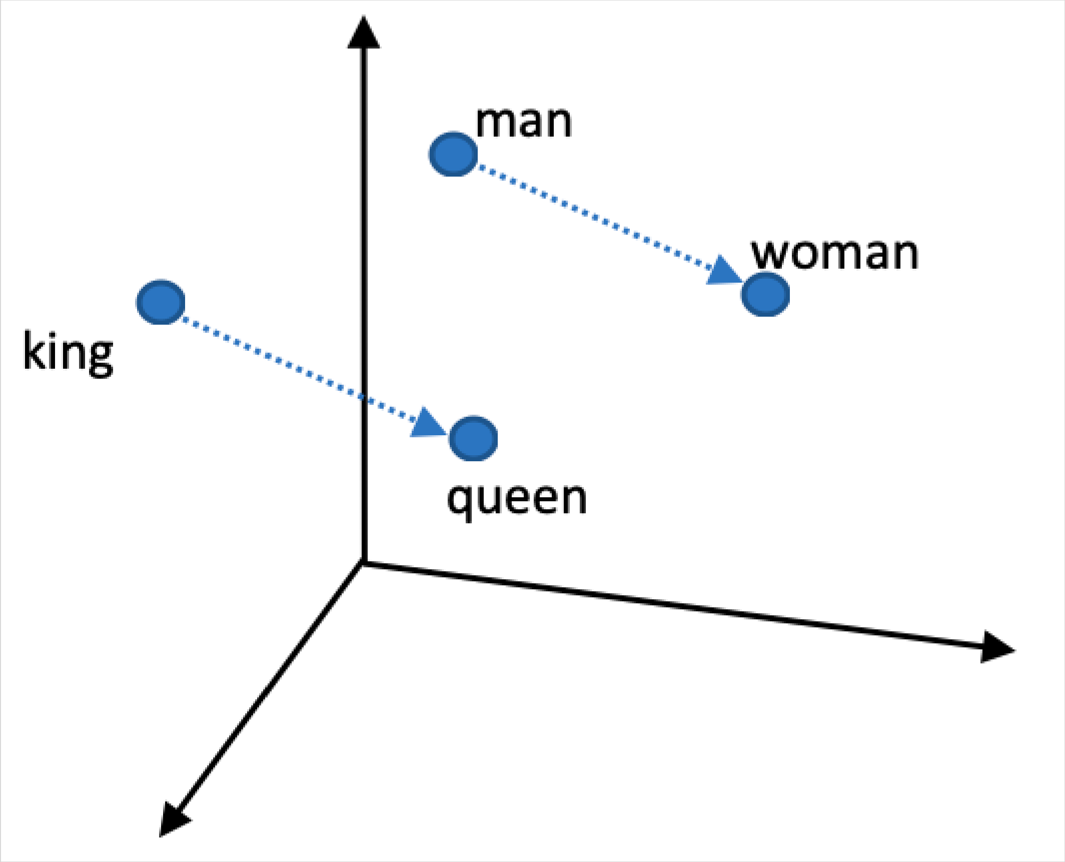
TEXT SIMILARITY

Nalani Schumacher

Background

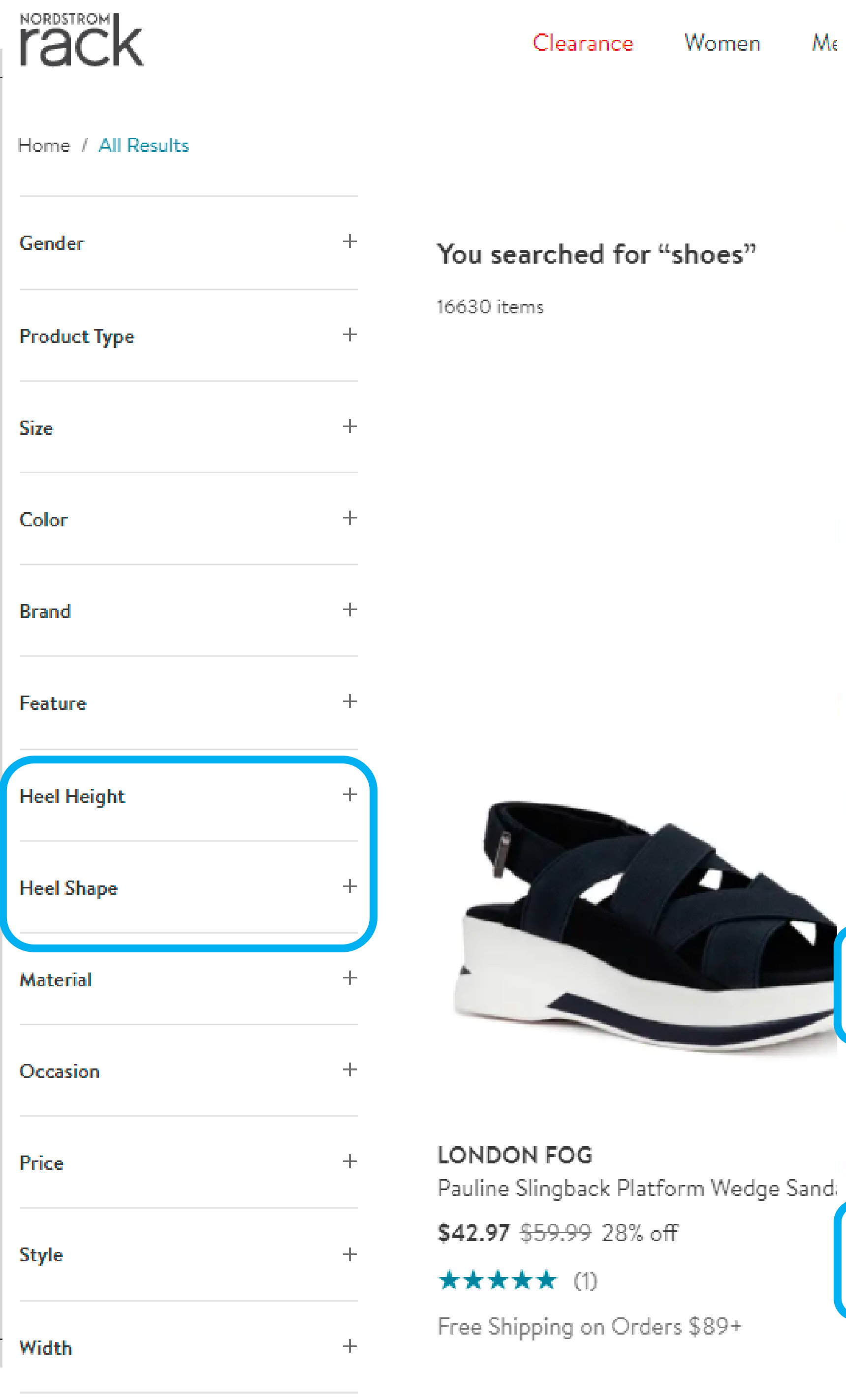
- Large natural language models with contextual, semantic word embeddings allow for text comparison
- Text similarity ranking
  - Words
  - Short Phrases
- Text pre-processing
  - Spelling
  - Profanity



Use Cases

Given a query

- Get related key words and phrases



NORDSTROM rack

Clearance Women Me

Home / All Results

Gender +

Product Type +

Size +

Color +

Brand +

Feature +

Heel Height +

Heel Shape +

Material +

Occasion +

Price +

Style +

Width +

You searched for “shoes”

16630 items

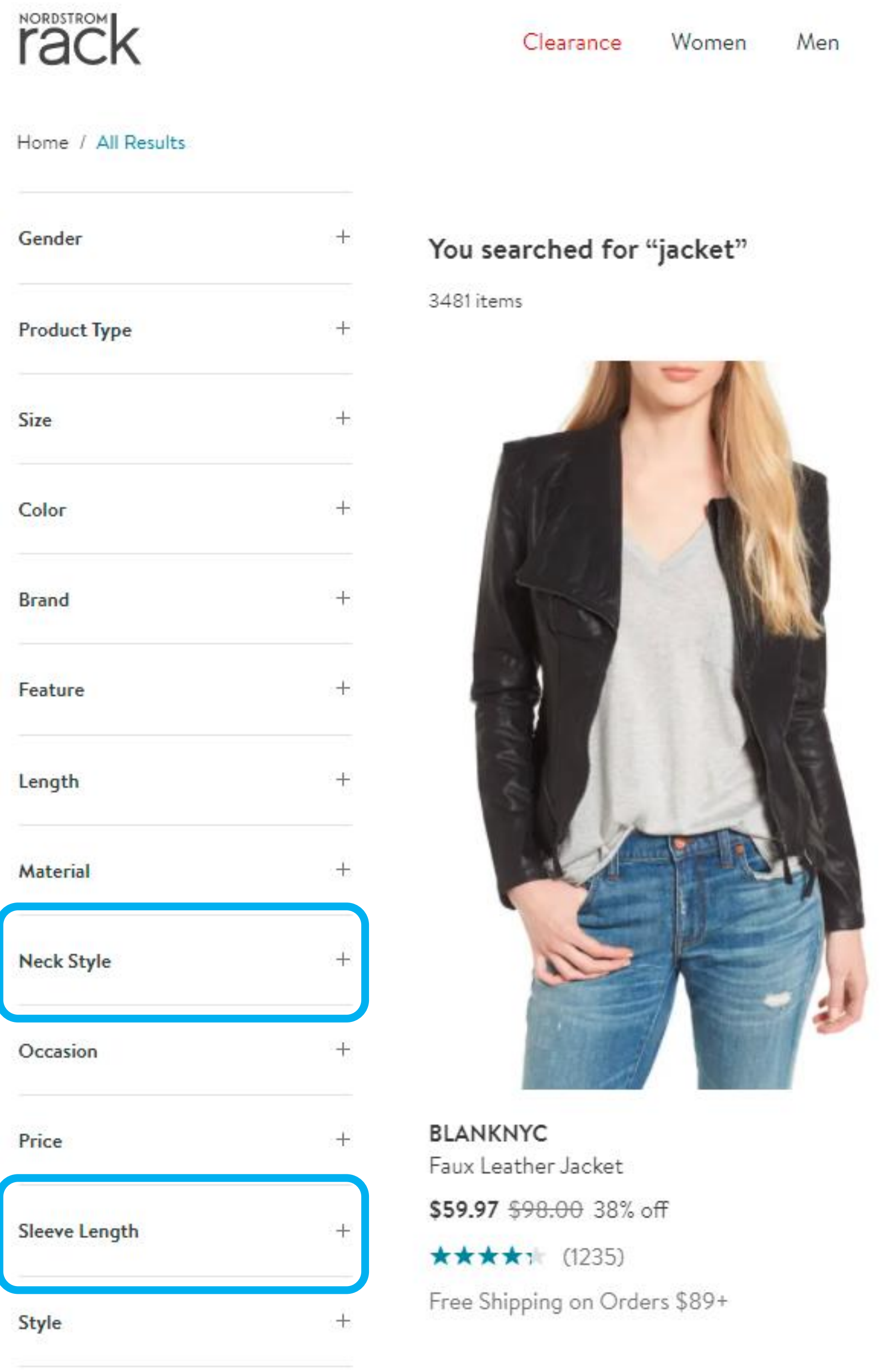
LONDON FOG

Pauline Slingback Platform Wedge Sand

$42.97 ~~$59.99~~ 28% off

★★★★★ (1)

Free Shipping on Orders $89+



NORDSTROM rack

Clearance Women Men

Home / All Results

Gender +

Product Type +

Size +

Color +

Brand +

Feature +

Length +

Material +

Neck Style +

Occasion +

Price +

Sleeve Length +

Style +

You searched for “jacket”

3481 items

BLANKNYC

Faux Leather Jacket

$59.97 ~~$98.00~~ 38% off

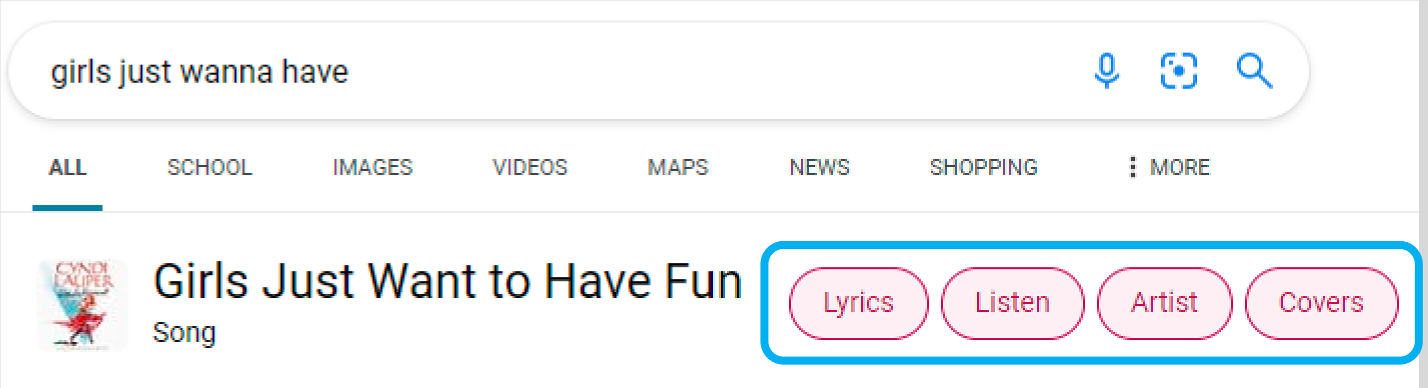
★★★★☆ (1235)

Free Shipping on Orders $89+

Use Cases

Given a query

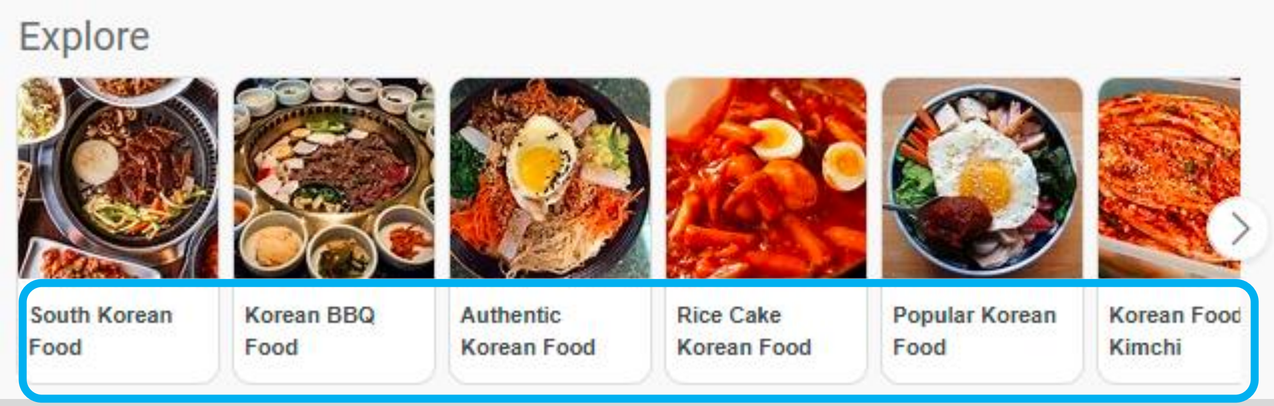
- Get related key words and phrases



girls just wanna have

Girls Just Want to Have Fun

Song



Explore

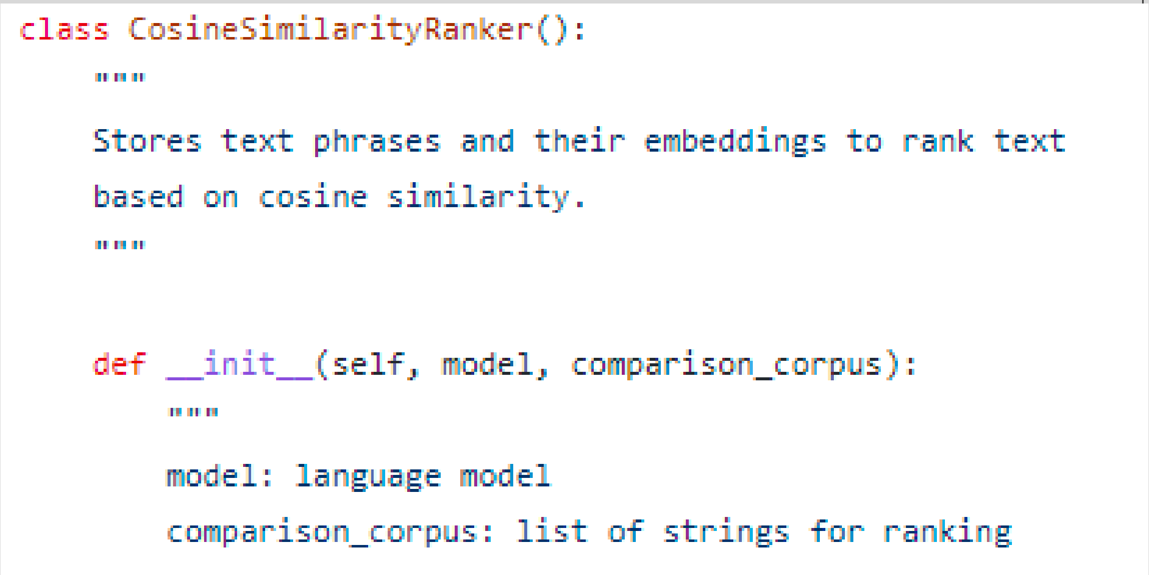
DEMO

Design

- Textsimilarity package has 3 modules separated by functionality and purpose
    - Cleaning text
        - Class to load supporting dictionaries and instances only once during initialization
        - Private method for calculating the Jaccard distance as a helper for spelling correction
    - Loading a language model
        - One class per model to load
        - Private methods for tokenizing data and retrieving text embeddings
    - Ranking a text corpus
        - One class per ranker
        - Private methods for calculating cosine similarity and creating a dictionary of embeddings

Design

- The ranker objects in the rankers module take in a language model as an input parameter.
  - Allows language model complexity to remain abstracted from the user.
  - Calls private methods in the model object to tokenize the data and get embeddings.
  - To rank the corpus based on a given text, the user just needs to call rank_on_similarity() passing in the target text.



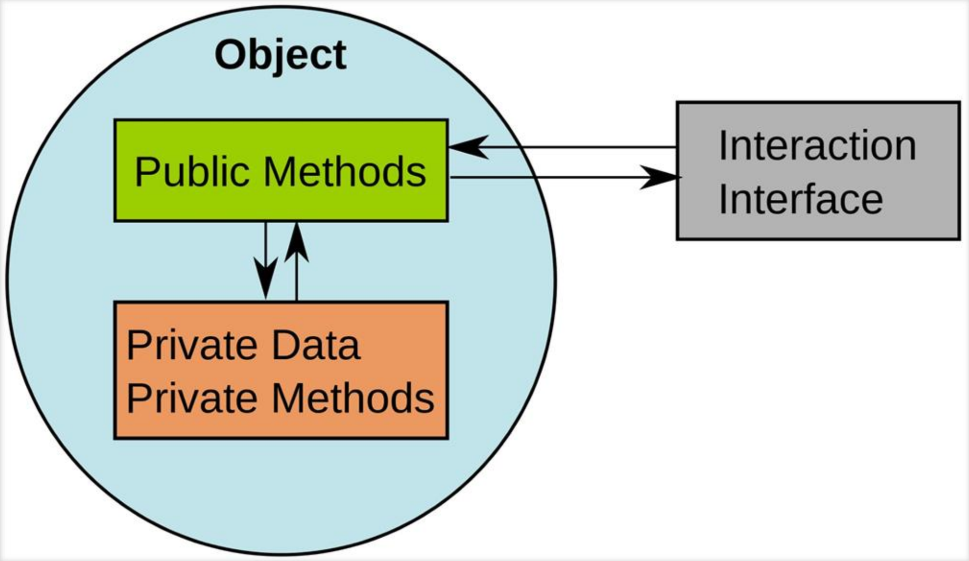
```
class CosineSimilarityRanker():
    """
    Stores text phrases and their embeddings to rank text
    based on cosine similarity.
    """

    def __init__(self, model, comparison_corpus):
        """
        model: language model
        comparison_corpus: list of strings for ranking
```

Design

- The CleanText object allows users to have better ranked results by not including profane text and by correcting spelling errors; however, if for instance a user knew their text was already clean then they could choose to save time and skip this step. Then they could continue to use the text_models and rankers modules as normal.

PROJECT STRUCTURE

Lessons Learned



- Tools
  - Travis-CI
    - Run all tests & generate coverage report
  - Unittest
    - Set up / tear down methods
- Design
  - Verb or action prefix on methods for clarity: “calculate”, “get”, “tokenize”, “rank”
    - Easier to understand, clearer expectations
  - Separating functionality: modules, classes, methods
    - Easily extendable and modifiable
  - Abstraction, information hiding
    - Easier to use

Future Work

In the package:

- Cross-modality expansion such as with text to image similarity

In the clean text module and text models module:

- Add support for other languages besides English

In text models module:

- Add more natural language models

In the rankers module:

- Add other rankers