The Use of NLP to Solve Problems

Annie Flippo

11/2/2016

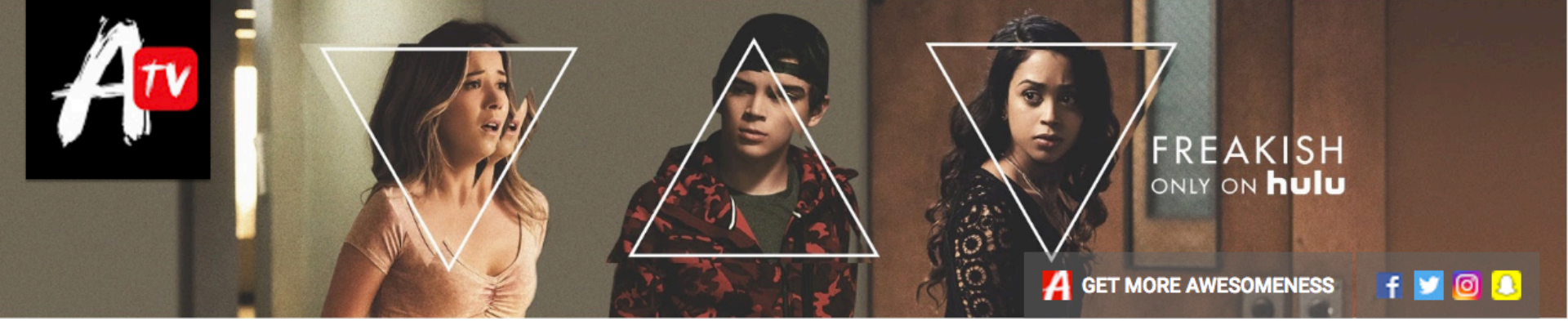
Who am I?

Annie Flippo

Sr. Data Scientist

AwesomenessTV / Dreamworks Animation SKG

Slides at bit.ly/acflippo-nlp

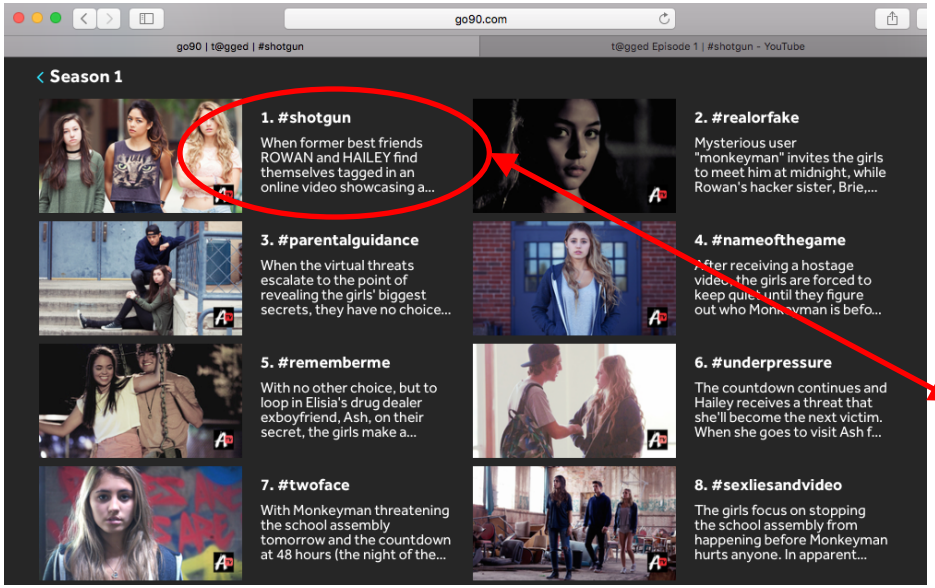


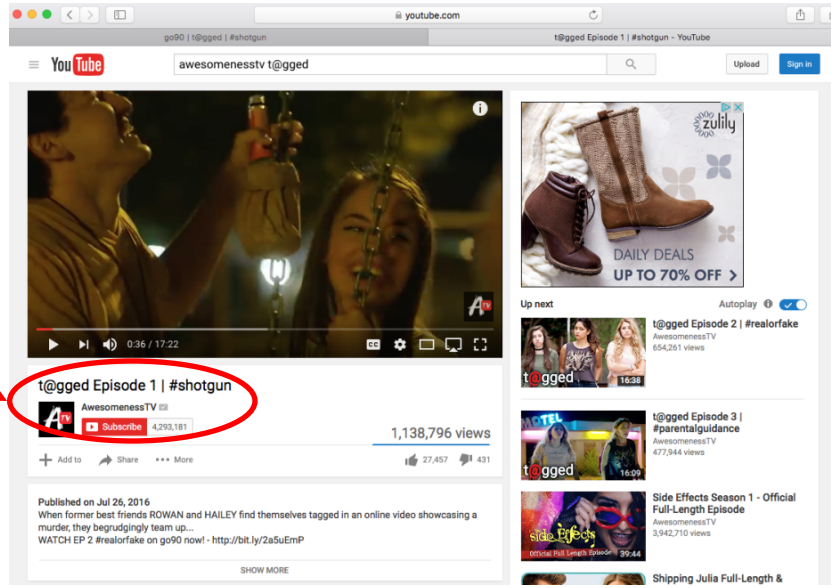
Who is AwesomenessTV?

We’re digital content provider for platforms including Hulu, Netflix, Roku, Verizon & YouTube.

Business Problem

Many systems managing videos on different platforms





Goals

Develop a method to identify same or similar assets across systems:

- Show asset relationship
- Generate unique id for in-house apps

Why use NLP?

Top goals for Natural Language Processing are:

1. Document Similarity (search engine query)

2. Topic Modeling (Twitter/Blog Analysis)

3. Sentiment Analysis (movie or restaurant reviews)

Data Processing

Why perform text processing?

- To rid of messiness of free-from text
- To group words with the same meaning
- Convert text to numeric features
- Model on equivalent numeric features

Data Processing

Titles and descriptions get scrubbed

- Remove punctuation, non-ascii, carriage returns
- Remove stop words (i.e. it, this, and, that)
- Stemming
- Lemmatize
- Tokenize
- Vectorize

Stemming

Reduce to the root of the word

Provision, providing, provider, provided

=> provid

Argue, argues, arguing, argued => argu

Lemmatize

Retrieve the linguistic root of the word

Walk, walking, walked => walk

Is, am, are => be

Begin, began, begun => begin

*Nouns and verbs are lemmatized differently.

Tokenize

Count distinct words from a corpus

“The quick brown fox jumped over the lazy dogs”

becomes

[‘the’, ‘quick’, ‘brown’, ‘fox’, ‘jump’, ‘over’, ‘lazy’, ‘dog’]

Vectorize

Count occurrences from distinct word vector.

“The quick brown fox jumped over the lazy dogs”

Tokenized to:

[‘the’, ‘quick’, ‘brown’, ‘fox’, ‘jump’, ‘over’, ‘lazy’, ‘dog’]

Vectorized to:

[2, 1, 1, 1, 1, 1, 1, 1]

Bag-of-Words Comparison

Doc 1: “The quick brown fox jumped over the lazy dogs”

Doc 2: “The quick fox ran away from the dog”

After processing, the corpus attribute vector is:

[‘quick’, ‘brown’, ‘fox’, ‘jump’, ‘over’, ‘lazy’,
‘dog’, ‘run’, ‘away’, ‘from’]

Two documents vectorize to:

Doc 1: [1, 1, 1, 1, 1, 1, 1, 0, 0, 0]

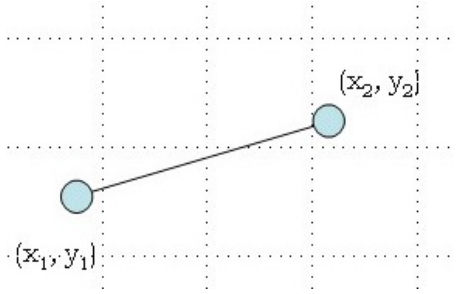
Doc 2: [1, 0, 1, 0, 0, 0, 1, 1, 1, 1]

Sentences are transformed into numeric vectors!

Similarity Measure

Cosine similarity calculates how close 2 numeric vectors are which is like the distance measure between 2 points.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$



This problem has just reduced to simple matrix algebra.

Bi-Gram Comparison

Due to the same words used across our videos, the bag-of-words similarity resulted high false positive matches.

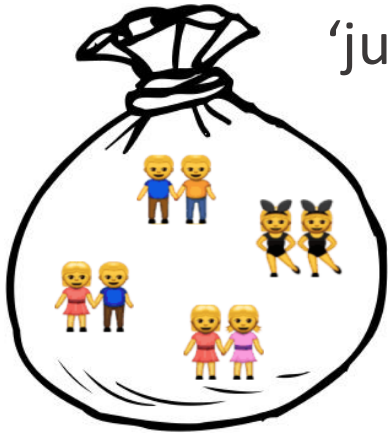
The solution is to use a Bi-Gram algorithm where 2 consecutive words are extracted as one feature:

"The quick brown fox jumped over the lazy dogs"

becomes:

['the quick', 'quick brown', 'brown fox', 'fox jump',

'jump over', 'over the', 'the lazy', 'lazy dog']



Limitations

Certain phrases such as “Behind the scenes” are found frequently. This creates an artificially high similarity score even if the videos are dissimilar.

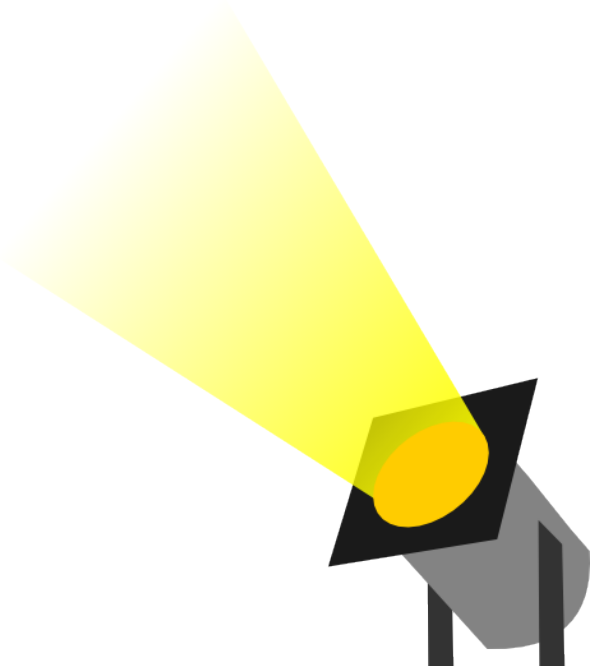
Possible solutions:

- Perform more custom data scrubbing
- Double-check by matching duration of videos
- Have the matches verified by a human

Conclusion

I use Natural Language Processing to:

1. Identify similar videos across platforms
2. Tie assets together where some are identical videos while others are derived videos (such as trailers or promos).



Slides and code are available at bit.ly/acflippo-nlp

Thank You!

Annie Flippo

@ACflippo