Annotation and Feature Engineering

Data Science: Jordan Boyd-Graber
University of Maryland
HOUSES, SPOILERS, AND TRIVIA

TV Tropes

- Social media site
- Catalog of “tropes”
- Functionally like Wikipedia, but . . .
  - Less formal
  - No notability requirement
  - Focused on popular culture

Absent-Minded Professor

- “Doc” Emmett Brown from Back to the Future.
- The drunk mathematician in Strangers on a Train becomes a plot point, because of his forgetfulness, Guy is suspected of a murder he didn’t commit.
- The Muppet Show: Dr. Bunsen Honeydew.

Spoilers

- What makes neat is that the dataset is annotated by users for **spoilers**.
- A spoiler: “A published piece of information that divulges a surprise, such as a plot twist in a movie.”

Spoiler

- Han Solo arriving just in time to save Luke from Vader and buy Luke the vital seconds needed to send the proton torpedos into the Death Star’s thermal exhaust port.
- Leia, after finding out that despite her (feigned) cooperation, Tarkin intends to destroy Alderaan anyway.
- Luke rushes to the farm, only to find it already raided and his relatives dead harkens to an equally distressing scene in The Searchers.

Not a spoiler

- Diving into the garbage chute gets them out of the firefight, but the droids have to save them from the compacter.
- They do some pretty evil things with that Death Star, but we never hear much of how they affect the rest of the Galaxy. A deleted scene between Luke and Biggs explores this somewhat.
- Luke enters Leia’s cell in a Stormtrooper uniform, and she calmly starts some banter.

The dataset

- Downloaded the pages associated with a **show**. Took complete sentences from the text and split them into ones with spoilers and those without
- Created a balanced dataset (50% spoilers, 50% not)
- Split into training, development, and test **shows**

The dataset

- Downloaded the pages associated with a **show**. Took complete sentences from the text and split them into ones with spoilers and those without
- Created a balanced dataset (50% spoilers, 50% not)
- Split into training, development, and test **shows**
  - Why is this important?

The dataset

- Downloaded the pages associated with a **show**. Took complete sentences from the text and split them into ones with spoilers and those without
- Created a balanced dataset (50% spoilers, 50% not)
- Split into training, development, and test **shows**
  - Why is this important?
- I'll show results using SVM; similar results apply to other classifiers

Step 1: The obvious

- Take every sentence, and split on on-characters.
- Input: “These aren’t the droids you’re looking for.”

Step 1: The obvious

- Take every sentence, and split on on-characters.
- Input: “These aren’t the droids you’re looking for.”

Features

These:1 aren:1 t:1 the:1 droids:1 you:1 re:1 looking:1 for:1

	False	True
False	56	34
True	583	605

Accuracy: 0.517

Step 1: The obvious

- Take every sentence, and split on on-characters.
- Input: “These aren’t the droids you’re looking for.”

Features

These:1 aren:1 t:1 the:1 droids:1 you:1 re:1 looking:1 for:1

What’s wrong with this?

	False	True
False	56	34
True	583	605

Accuracy: 0.517

Step 2: Normalization

- Normalize the words
  - Lowercase everything
  - Stem the words (not always a good idea!)
- Input: “These aren’t the droids you’re looking for.”

Step 2: Normalization

- Normalize the words
  - Lowercase everything
  - Stem the words (not always a good idea!)
- Input: “These aren’t the droids you’re looking for.”

Features

these:1 are:1 t:1 the:1 droid:1 you:1 re:1 look:1 for:1

	False	True
False	52	27
True	587	612

Accuracy: 0.520

Step 3: Remove Usless Features

- Use a “stoplist”
- Remove features that appear in > 10% of observations (and aren’t correlated with label)
- Input: “These aren’t the droids you’re looking for.”

Step 3: Remove Usless Features

- Use a “stoplist”
- Remove features that appear in > 10% of observations (and aren’t correlated with label)
- Input: “These aren’t the droids you’re looking for.”

Features
droid:1 look:1

	False	True
False	59	20
True	578	621

Accuracy: 0.532

Step 4: Add Useful Features

- Use bigrams (“these_are”) instead of unigrams (“these”, “are”)
- Creates a lot of features!
- Input: “These aren’t the droids you’re looking for.”

Step 4: Add Useful Features

- Use bigrams (“these_are”) instead of unigrams (“these”, “are”)
- Creates a lot of features!
- Input: “These aren’t the droids you’re looking for.”

Features

these_are:1 aren_t:1 t_the:1
the_droids:1 you_re:1
re_looking:1 looking_for:1

	False	True
False	203	104
True	436	535

Accuracy: 0.578

Step 5: Prune (Again)

- Not all bigrams appear often
- SVM has to search a long time and might not get to the right answer
- Helps to prune features
- Input: “These aren’t the droids you’re looking for.”

Step 5: Prune (Again)

- Not all bigrams appear often
- SVM has to search a long time and might not get to the right answer
- Helps to prune features
- Input: “These aren’t the droids you’re looking for.”

Features

these_are:1 the_droids:1
re_looking:1 looking_for:1

	False	True
False	410	276
True	229	363

Accuracy: 0.605

How do you find new features?

- Make predictions on the development set.
- Look at contingency table; where are the errors?
- What do you miss?

How do you find new features?

- Make predictions on the development set.
- Look at contingency table; where are the errors?
- What do you miss? **Error analysis!**
- What feature would the classifier need to get this right?
- What features are confusing the classifier?
  - If it never appears in the development set, it isn't useful
  - If it doesn't appear often, it isn't useful

How do you know something is a good feature?

- Make a contingency table / scatter plot for that feature (should give you good information gain and be random)
- Throw it into your classifier (accuracy should improve)