A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

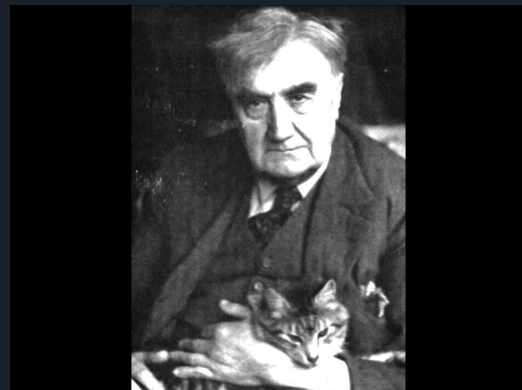
Kickstarter Document Classification

William Lane

Roadmap:

This presentation is structured to present my thought process and progression through this task:


- Quick Data Description
- First thoughts on how to approach the problem and why
- Creating a baseline/performance
- Iterative improvement
 - Successes
 - Failures
- Error analysis
- Further iteration
- Summary of results
- Ideas for other approaches to solve the same problem





The data

- I wanted to be able to incrementally improve on my solution informed by performance metrics.
 - Split the data into train/test splits: 80:20
 - 4878 training docs
 - 1220 testing docs
 - All data exploration, feature engineering choices, and model training was done on the training split
 - Test split was only used for generating performance metrics for models trained on training split
- The final model--the one used by my final classification script--is trained on both the training and test splits. Because mo' data is better data.



Approaching Document Classification of Kickstarter pitches:

After reading through some pitches, I formed a basic premise:

- Document classification by topic is driven by lexical content.
 - IE The content of what is talked about (lexicon) will correlate more strongly with the topic of discourse than aspects of syntax or semantic relations will.
 - (Of course this premise does not hold true for other NLP tasks such as information extraction, where tokens must be considered in the context of their relationships with other tokens).
- Bag of words: A suitable starting point.
 - Unigrams
 - Concatenated title, blurb, and main text body
- Chi2 feature selection
 - $k=15000$
- Linear SVC

Baseline:

Label	P	R	F1
Film/Video	78.0	80.0	79.0
Fashion	71.4	69.6	70.5
Crafts	50.0	46.9	48.4
Comics	78.1	78.1	78.1
Publishing	76.4	76.4	76.4
Art	61.8	48.8	54.5
Tech	63.8	66.6	65.2

Dance	100.0	75.0	85.7
Games	78.1	83.3	80.6
Photog	73.0	54.0	62.1
Journlsm	14.3	12.5	13.3
Food	74.5	85.5	79.1
Design	41.6	37.6	39.5
Music	84.4	93.1	88.5
Theater	60.0	60.0	60.0

Average f1: 65.39

Unigrams + bigrams + NER subbed text - stopwords + chi squared feat selection(k=15000)

Label	P	R	F1
Film/Video	81.9	84.5	83.1
Fashion	68.6	60.8	64.4
Crafts	41.7	30.6	35.2
Comics	81.8	84.4	83.1
Publishing	77.9	75.7	76.8
Art	60.2	54.6	57.3
Tech	63.6	67.4	65.5

Dance	92.3	75.0	82.8
Games	75.3	81.1	78.1
Photog	71.8	56	62.9
Journlsm	57.1	25.0	34.8
Food	60.6	82.9	70.0
Design	43.1	36.5	39.5
Music	85.6	90.3	87.9
Theater	65.2	60.0	62.5

Average f1: 65.59



An idea! (...which didn't pan out)

1. I know I need to find a way to highlight the meaningful words and phrases, and de-emphasise everything else. What if I retrieve a list of terms and phrases per text blob that are likely to be representative of the theme?
 - THE PROCESS:
 - Pull out the top N most frequent noun phrases from the text blog and create a bag-of-popular-phrases model.
 - THE RESULT:
 - Significantly smaller vectors. Too small in fact. In order to get NPs that are actually repeated more than once, K usually has to be < 15 or so, otherwise the list of singletons that follow are simply random NPs selected for no other reason than chance.
 - Performance was pretty bad. We do not speak of those number here.
 - WHY IT DIDN'T WORK:
 - Super oversimplified representation of a document.
 - Lost access to verbs and other tokens that may have had an impact on the decision boundary

Unigrams + TFIDF(maxdf=.5) - stopwords

Label	P	R	F1
Film/Video	84.1	81.9	83.0
Fashion	63.4	65.8	64.6
Crafts	54.8	34.7	42.5
Comics	100	68.8	81.5
Publishing	74.3	74.3	74.3
Art	64.8	53.5	58.6
Tech	50.5	68.9	58.3

Dance	93.8	93.8	93.8
Games	72.6	85.6	78.6
Photog	69.4	50.0	58.1
Journlsm	66.7	12.5	21.1
Food	75.0	79.5	77.2
Design	48.6	42.4	45.3
Music	85.0	93.8	89.2
Theater	80.0	64.0	71.1

Average f1: 66.48

Unigrams + TFIDF(maxdf=.25, max_features=15000) - stopwords

Label	P	R	F1
Film/Video	82.6	89.0	85.7
Fashion	77.5	78.4	78.0
Crafts	60.0	55.1	57.4
Comics	89.3	78.1	83.3
Publishing	79.1	78.6	78.9
Art	68.6	55.8	61.5
Tech	72.6	78.5	75.4

Dance	88.2	93.8	90.9
Games	80.4	91.1	85.4
Photog	70.1	58.0	63.7
Journlsm	50.0	25.0	33.3
Food	86.9	90.6	88.7
Design	58.8	58.8	58.8
Music	90.2	95.2	92.6
Theater	88.2	71.4	71.4

Average f1: 73.68

Unigrams + bigrams+ TFIDF(maxdf=.25, max_features=15000) - stopwords

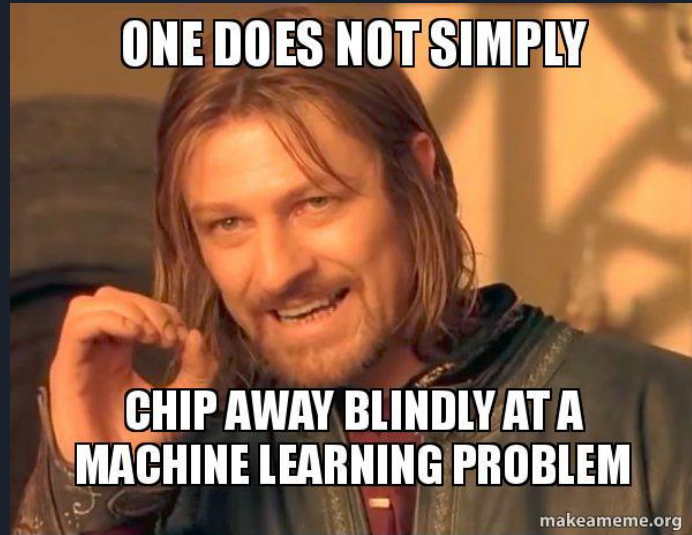
Label	P	R	F1
Film/Video	81.5	88.4	84.8
Fashion	78.5	78.5	78.5
Crafts	58.3	57.1	57.7
Comics	92.6	78.1	84.7
Publishing	78.7	76.4	77.5
Art	68.6	55.8	61.5
Tech	74.7	80.7	77.5

Dance	87.5	87.5	87.5
Games	80.0	88.9	84.2
Photog	74.4	64.0	68.8
Journlsm	66.7	25.0	36.4
Food	89.25	92.3	90.8
Design	58.0	60.0	59.0
Music	90.1	94.5	92.3
Theater	90.0	72.0	80.0

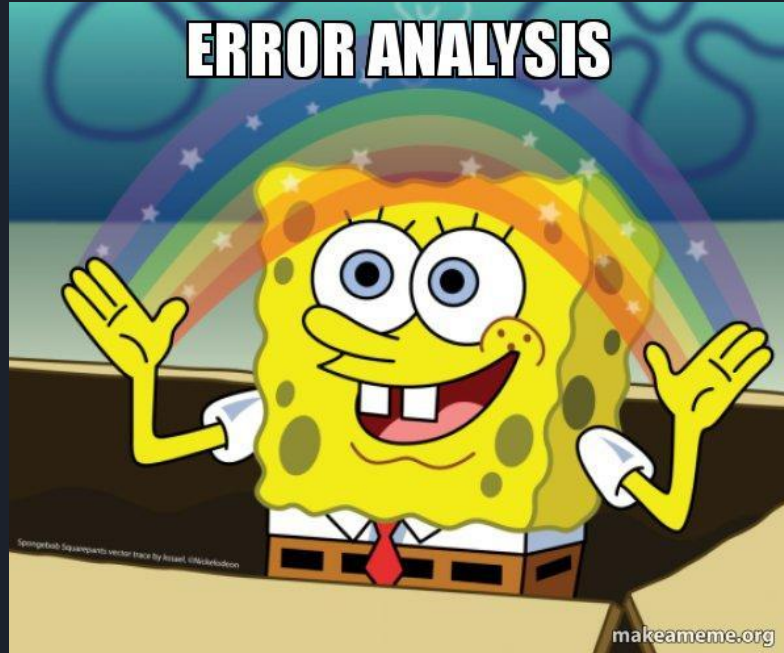
Average f1: 74.75

Taking a step back for a second

- I've improved on the baseline at this point by an average f-score of = 9.36%
- Gains have slowed, but some scores (I'm looking at you, Journalism) are still very low.
- Why?



Seems like it's about time for some good old-fashioned



Alright Journalism, what's your deal?

Journalism	P: 66.7	R: 25.0	F1: 36.4
------------	---------	---------	----------

- Test set contained 16 Journalism pitches:
 - 4 true positives
 - Launching a social network for news media,
 - Producing radio audio documentaries
 - Supporting an aspiring travel journalist
 - Producing a radio podcast.
 - 12 false negatives
 - 2 false positives





Alright Journalism, what's your deal?

Out of 16 gold-label
Journalism pitches:

Journalism	P: 66.7	R: 25.0	F1: 36.4
------------	---------	---------	----------

- 12 false negatives:
 - “Journalism” where the pitch focuses on ideas that vary wildly in theme: 8
 - Doc162: A man writes his pitch about how he was adopted and never met any of his siblings. He wants to find them.
 - Doc270: A man wants to start a blog about student life; parties, nightlife, parties, cool party stories, awesome bruuuh.
 - Doc575: Description of the Rio Grande, its ecology, local indigenous customs, etc
 - Doc852: Describes a high school soccer club
 - Human error: This isn't journalism at all: 2
 - Doc 732: “Fund me so I can go learn spanish and become a bilingual insurance adjuster”
 - Doc1125: “Fund me so i can go on youtube and recap the first five episodes of supernatural”
 - Light on keywords/keywords misspelled consistently: 2
 - Doc506: Man pitches a podcast called “Pudcast”. In my defense, he only uses the word ‘podcast’ 1 time, the rest of the time he says ‘the Pudcast’.
 - Doc935: Short description of a proposed paperback magazine. Light on keywords.



Alright Journalism, what's your deal?

Journalism	P: 66.7	R: 25.0	F1: 36.4
------------	---------	---------	----------

- Test set contained 16 Journalism pitches:
 - 4 true positives
 - 12 false negatives
 - 2 false positives
 - Lexical/Thematic overlap (These kinda “seem” Journalist-y, even to a human):
 - Doc89: A pitch to set up a website where students can **interview** professionals in their field to get mentorship and career advice. (Gold==Publishing))
 - Doc661: Pitches a 1-800 call-in line inspired by a previous call-in **radio program**. Much of the content here are summaries of **radio interviews** conducted by the Kickstarter author. (Gold == Film&Video)



Takeaways from Journalism error analysis

Journalism	P: 66.7	R: 25.0	F1: 36.4
------------	---------	---------	----------

- Journalism is tough to classify on a lexical basis, because the subject of the journalistic endeavor can vary wildly and overlap with other themes:
 - A pitch proposes broadcasting video game e-sport coverage on a website. Topics this overlaps with:
 - Technology
 - Games
 - Journalism
 - This creates a very noisy feature space
- Human Error is a factor to consider from data scraped off the web:
 - People may simply mis-categorize their pitch
 - How to mitigate?
 - Maybe have human annotators comb through data to verify and re-assign labels as necessary
 - Get interrater agreement to establish the “max ceiling” for our machine predictions



Further Error Analysis Confirms trends seen in Journalism Analysis

The most common sources of error across the board:

- Significant lexical overlap across some topics:
 - A dating service for gamers that gamefies dating through a new app built by game developers. Technology or Games?
 - A cooking show broadcasting recipes from around the world. Food or Journalism?
 - A Halloween digital shadowbox wall art device. Crafts? Art? Technology? Spooky?
 - Man designs and produces playing cards. Games or Design?
 - And so many more!
- People misclassifying projects -->

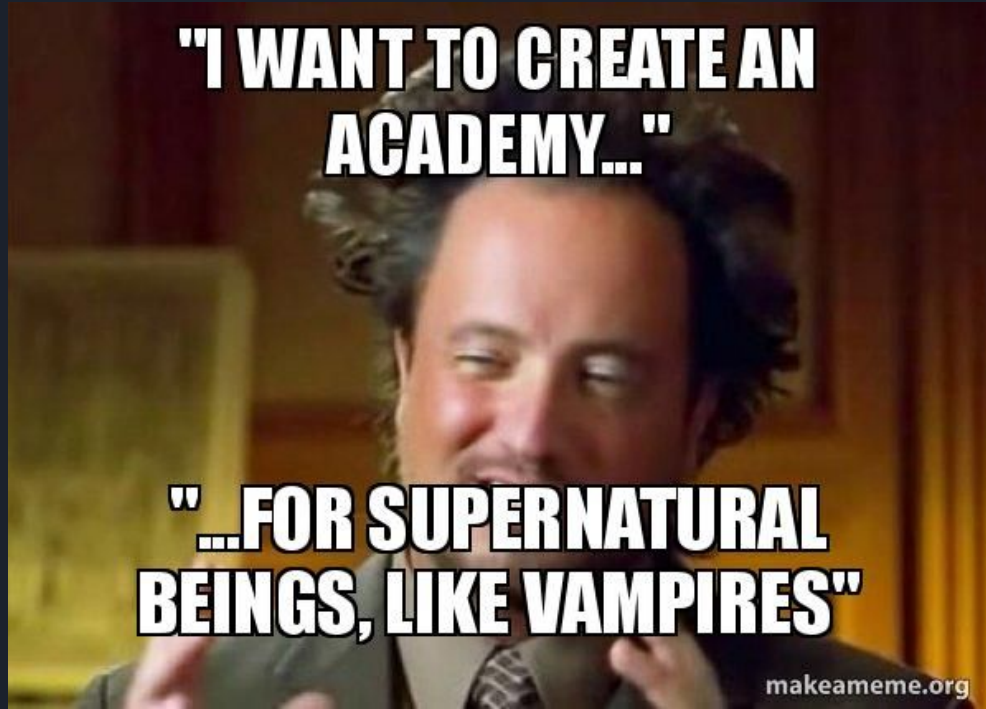
People Misclassifying their Projects:



People Misclassifying their Projects:



People Misclassifying their Projects:




This was categorized as
Journalism, obviously...

People Misclassifying their Projects:



Also journalism.



An Attempt to collapse terms into semantic classes

Performing Arts:

```
def sub_performing_arts_terms(text):  
    regexes = " performing arts ", " performance ", " dance ", " spoken word ", " opera "
```

Crafts:

```
def sub_craft_terms(text):  
    regexes = [" quilting ", " embroidery ", " crafts ", " mosaic ",  
               "porcelain", " glassware ", " pottery ", " tatting ", " rug-making ", " stained glass ", " wrought iron ",  
               " glass making ", " glassblowing ", " glass etching ", " weaving ", " knitting ", " crochet ", " felting ",  
               " floral design ", " bouquet ", " leather crafting ", " carving ", " wood carving ", " scrapbooking ", " calligraphy ",  
               " origami ", " decoupage ", " bookbinding ", " embossing ", " calligraphy ", " carpentry ", " upholstery ", " woodworking",  
               " intarsia ", " stonemason ", " jewellery ", " silversmith ", " handicraft ", " charms "  
    ]
```

Modes of media delivery:

```
def sub_journalism_terms(text):  
    regexes = " radio ", " magazine ", " documentary ", " newsletter ", " web magazine ", " travel blog ", " newsletters ", " interview ",  
               "interviews ", " online publication ", " online publications ", " essays ", " essay ", " social media ", " photo journalism ", " public  
               television ", " public broadcasting ", " television news ", " audio journalism ", " audio documentaries "
```

Results of the attempt to collapse words into semantic classes:

Label	P	R	F1
Film/Video	80.5	87.7	84.0
Fashion	78.5	78.5	78.5
Crafts	61.7	59.2	60.4
Comics	96.0	75.0	84.2
Publishing	78.0	78.6	78.3
Art	72.5	58.1	64.5
Tech	74.5	80.0	77.14

Dance	93.3	87.5	90.3
Games	79.2	88.9	83.8
Photog	77.5	62.0	68.9
Journlsm	80.0	25.0	38.1
Food	89.3	92.3	90.8
Design	58.0	60.0	59.0
Music	89.7	95.9	92.7
Theater	90.0	72.0	80.0

Average f1: 75.4



Concerns about collapsing terms into semantic classes:

- Overfitting to the dev set:
 - If I just build my dicts on terms I see in the FN and FP's of my dev data split, I'll obviously see my scores go up on that dev data split.
 - This doesn't necessarily mean my model performs better on unknown instances
 - How to mitigate?
 - For ideas, I did browse the data I generated for error analysis. But I also wanted to make sure my lists were exhaustive and contained other terms not observed in the dev split.
 - Consulted wikipedia/google for more complete lists of “crafts”, “performing arts”, “modes of journalistic delivery”
 - Of course, all these lists could/should be expanded further
- It's a long manual process to curate custom dictionaries/ontologies:
 - It's possible resources like this exist and can be adopted: it's always smart to check that out before jumping into building out your own...



One more idea:

- Error analysis showed that text blobs of varying lengths were very noisy, especially as you progress further into the narrative. I wonder if the tendency holds that the most summative information is expressed at the beginning of the pitch?
 - THE PROCESS:
 - For each pitch, just take the title, the blurb, and the first 20 sentences of the pitch, and pass that through the pipeline you've built up so far.
 - THE RESULT:
 - Started at 20, got avg f1 of 73.5
 - Reduced to 10, got avg f1 of 72.4
 - Reduced to 5, got avg f1 of 68.5
 - Increased to 40, got avg f1 of 73.1
 - WHY IT DIDN'T WORK:
 - While the rule that the most summative and attention-grabbing information in a text is found in the first couple sentences may hold for formal texts like newspapers and articles, pitches written by amateur writers of all ages and experiences are not guaranteed to hold to that principle, no matter how ideal it may be to format a pitch that way.



Summary

- I improved the average F1 score across all categories by +9.94% over the baseline with:
 - Unigrams
 - Bigrams
 - TFIDF with a `max_df=.25` to trim away words occurring in more than 25% of texts
 - Filtering the feature vectors down from ~43,000 dimensions to the top 15,000 most important features according to TFIDF ranking
 - Collapsing terms into semantic classes
- I tried a number of things that didn't work and learned from the experience
- I performed an in-depth error analysis and learned a lot about the data:
 - The “gold labels” aren't necessarily gold: human error in initial categorization affects the machine learning algorithm's ability to capitalize on patterns
 - My initial premise that text classification was a simple matter of associating lexicon with labels was challenged:
 - I still think lexical features are probably the most predictive of thematic category, but it's not a “simple” matter by any means, but the problem of lexical overlap across categories is non-trivial.
 - Discourse level features are unlikely to be helpful given the lack of formality and uniformity of writing structure inevitable on a site that lets anyone write a pitch in whatever way they want



Ideas for future approaches to this problem

- LDA to discover multiple topics per document, possibly create some logic that favors cross-topical categories like Journalism even if other topics outrank it.
 - Eg: DocumentA:
 - 62% Games (computers, games, tournament, etc)
 - 38% Journalism (interviews, broadcast, report, etc)
 - In this case we know Journalism tends to be cross-topical and select it as the primary topic despite its comparatively weaker representation in the full document
- Word embeddings to represent and compare document semantics
 - more dense representation than the incredibly sparse BOW models
 - 100-300 dimensions usually vs. vectors that can easily exceed 50,000.
 - While BOW can be trimmed down by feature selection, word embeddings are more dense, preserving the semantic relation between all words without trimming the original vocabulary
 - Downside: Word embedding representations assume meaning is compositional average of all words in document.
 - Real meaning doesn't work that way in language, but for topic classification the generalization may work alright.