Classifying Medical Documents

General Assembly Data Science Part Time (Jan-Mar 2017) – Final Project

Sierra Costanza

Introduction

- **Project Objective:** Medical practices handle a large amount of documents for each of their patients (many coming from different sources) that often require manual work of administrators to classify, compile and file. My friend Brian is developing a startup called WaitingRoomApp, an app that helps automate several aspects of the waiting room in medical practices. This project helps with the task of automating the process of classifying the documents that the practices receive, and could eventually serve as one of WaitingRoomApp's product offerings and save its customers a considerable amount of time, money and resources.
- **Data Source:** 4 types of medical document pdfs queried using the Microsoft Bing Web Search API:
  1. medical release forms
  2. informed consent forms
  3. patient intake forms
  4. "catch all" bucket for other general medical forms that do not fit those 3 types

  Target variable = document type

  Built models using the data at both the <u>document level</u> and <u>page level</u>

EDA and Feature Engineering

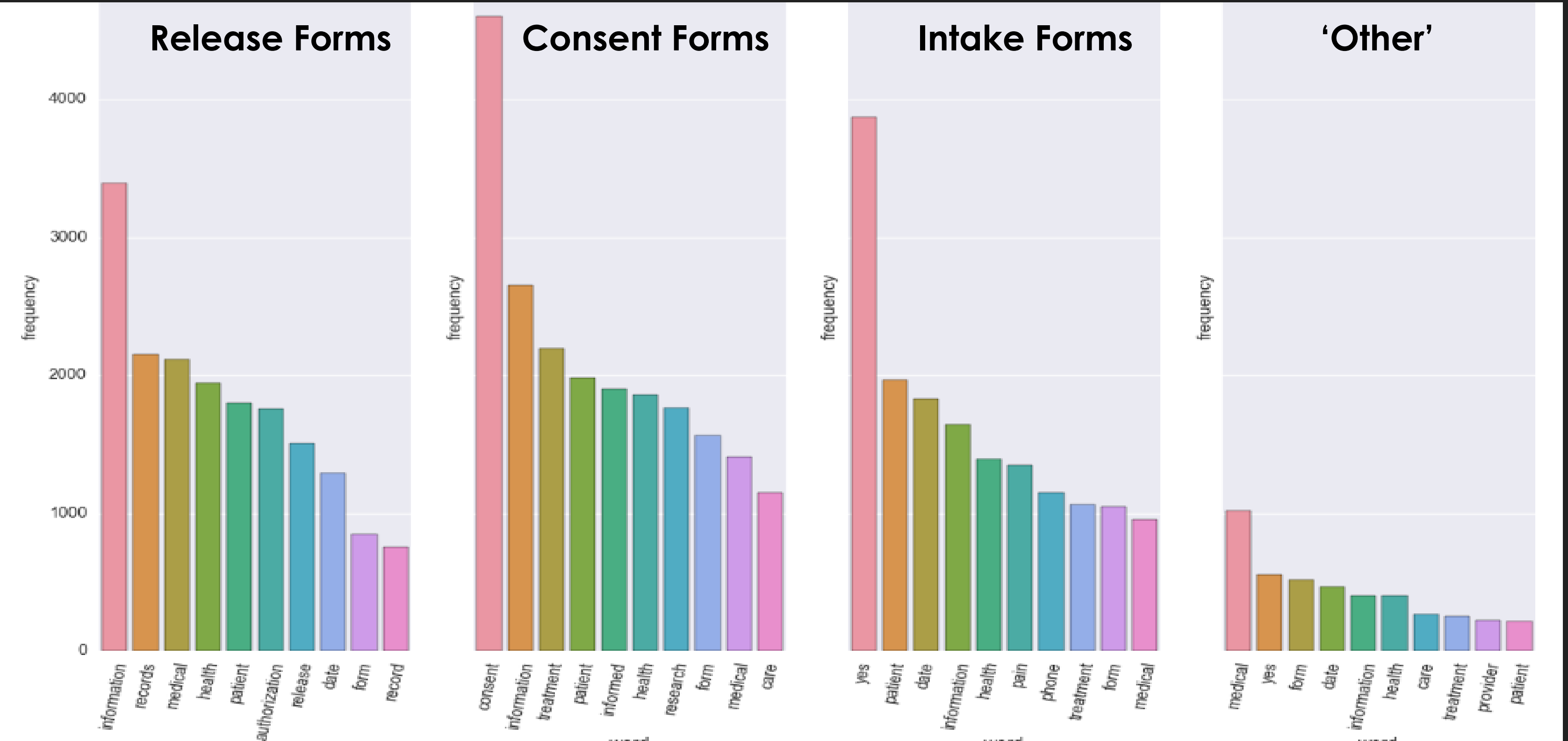
Class Distribution

Data	Release Forms	Consent Forms	Intake Forms	'Other'
Document Level	306 (23%)	467 (35%)	488 (36%)	87 (6%)
	Total: 1,348			
Page Level	601 (12%)	2,135 (41%)	2,170 (42%)	246 (5%)
	Total: 5,152			

Removed Duplicates →

Data	Release Forms	Consent Forms	Intake Forms	'Other'
Document Level	**296 (31%)**	**278 (29%)**	**295 (31%)**	**82 (9%)**
	Total: 951 | # words in corpus: 38,511			
Page Level	**578 (16%)**	**1,510 (42%)**	**1,305 (36%)**	**237 (7%)**
	Total: 3,630 | # words in corpus: 37,712			

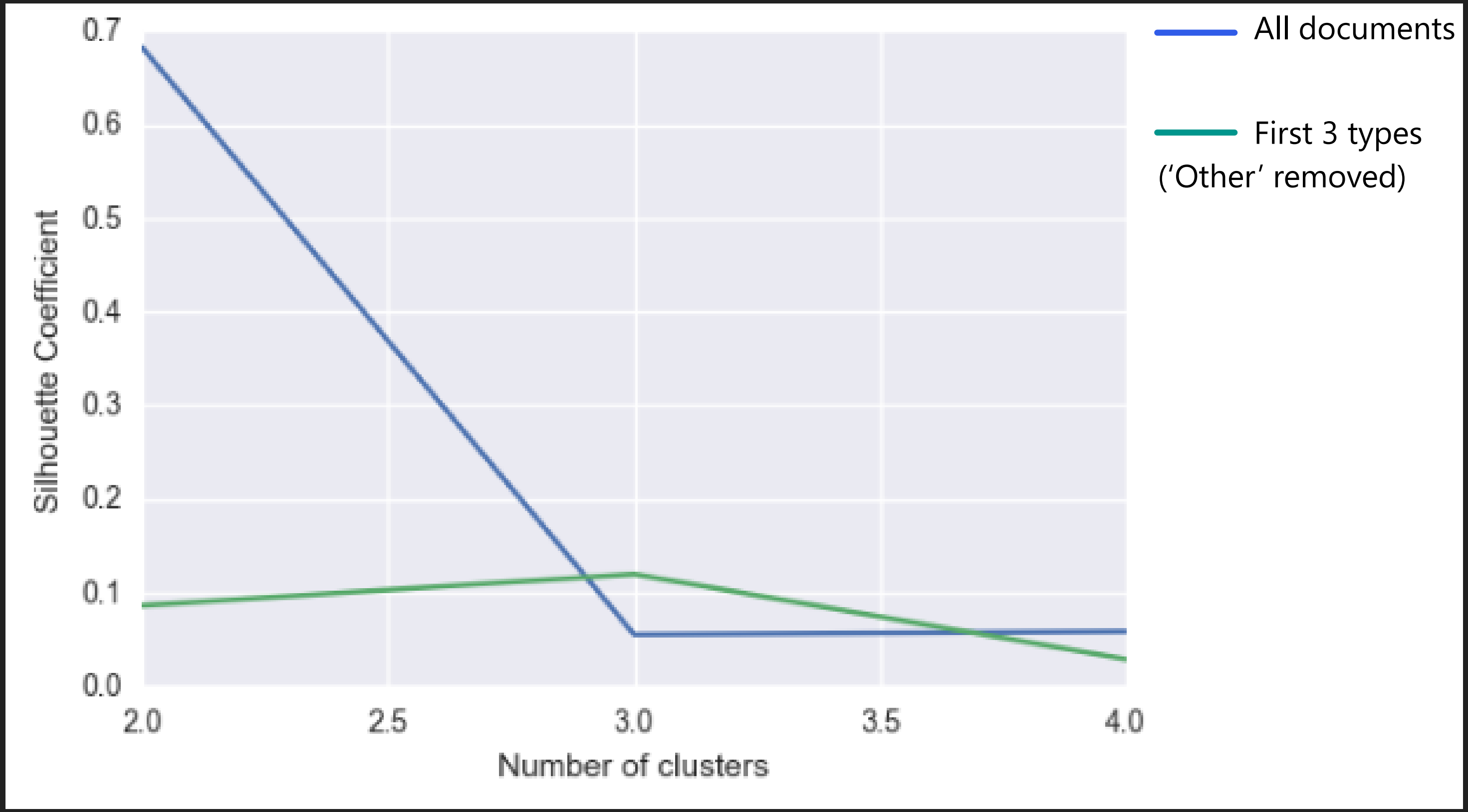
Most Frequent Words per Class



EDA and Feature Engineering

Clustering – is there inherent structure to the data without using the class labels?

Clustering performed on (scaled) Tf-Idf matrix for document level data (English stop words, 10K max features)



- Presence of 'Other' type makes 4 distinct document types hard to detect when all documents are included in clustering. This may (and did!) pose a problem when modeling
- When 'Other' is removed, 3 clusters are optimal and detected (although fairly low Silhouette Coefficient)

Modeling

- Split data into 75% train, 25% test
- Used Tf-Idf matrix (Just Tf performed similarly)
- Also tried using stems, but performed similarly/slightly worse in all cases

1. Multinomial Naive Bayes – Didn't work well for 4th class

Parameters Tuned:

- Stop words (None, English), Max features, N-grams, Min df
- NB: Fit prior (True/False), class prior (None, all equal)
- Used Grid Search with 4-fold cross validation on training data
- Also tried modeling using just the first 3 classes, taking the best model and *manually* classifying the 4th class with probability thresholds, but didn't perform as well as other algorithms below

2. RandomForest – Performed best Document Level data, worse on average than SVM

Parameters Tuned:

- Max features, N-grams, Min df
- RandomForest: Class weight, # estimators, max features, min samples split/leaf, criterion
- Used Grid Search with 3-fold cross validation on training data

3. Linear SVM ('hinge' loss function) with SGD

Parameters Tuned:

- Stop words (None, English), Max features, N-grams, Min df
- SGD: Class weight, alpha, # iterations
- Used Grid Search with 4-fold cross validation on training data

Final Models:

DOCUMENT LEVEL DATA - RandomForest

- Stop words = English, max features = 30,000
  N-grams = (1,3), min df = 3
- Class weight = Balanced
- # estimators = **10 (!)**, max features = 3,000, min samples leaf/split = 3, max depth=None, criterion = entropy

PAGE LEVEL DATA – Linear SVM with SGD

- Stop words = English, max features = 32,000
  N-grams = (1,3), min df = 1
- Class weight = Balanced
- SGD: alpha = 0.0003, # iterations = 13

Results

Multinomial Naive Bayes – Didn't work well for 4th class

DOCUMENT LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	69	1	0	0
2	2	68	6	0
3	2	3	68	0
4	7	3	9	0

Accuracy
86.1%

Precision
79%

Recall
86%

F1
83%

PAGE LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	145	6	3	0
2	11	347	13	0
3	19	26	295	0
4	8	6	18	11

Accuracy
87.9%

Precision
89%

Recall
88%

F1
87%

RandomForest

DOCUMENT LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	68	1	0	1
2	1	72	2	1
3	0	3	69	4
4	2	1	1	15

Accuracy
94.1%

Precision
94%

Recall
94%

F1
94%

Linear SVM with SGD

PAGE LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	144	3	1	6
2	1	357	12	1
3	5	15	320	0
4	0	4	5	34

Accuracy
94.2%

Precision
94%

Recall
94%

F1
94%

Extension: Tensorflow

Implemented a Convolutional Neural Network on Words (Tf matrix)
Shown to work for text classification

- Adapted from tensorflow open source Github repo example; uses tf.contrib.learn API
- LOTS of parameters to tune – *still more research to do here*
  - Examples: number of filters, filter shapes, pooling window and stride, number of layers, optimizer & learning rate (SGD, 'Adam' (used here)), number of steps…

Results So Far

DOCUMENT LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	56	1	2	11
2	3	58	3	12
3	0	3	56	14
4	3	3	3	10

Accuracy
75.6%

Precision
84%

Recall
76%

F1
79%

PAGE LEVEL DATA

Confusion Matrix (test data)

	1	2	3	4
1	113	30	10	1
2	53	311	4	3
3	73	10	257	0
4	22	9	9	3

Accuracy
75.3%

Precision
79%

Recall
75%

F1
76%

Next Steps

- **Model with more documents and for more medical document classes**

- **Image Classification:** *Just* a text classifier might not be sufficient for all potential use cases; for instance, medical practices could receive faxes or photocopies of documents that are rotated and the text might not be extractable. A next phase of the project would be to implement image classification using Convolutional Neural Networks with the document page inputs as images instead of text from pdfs.

- **Product Integration:** Turning the model into a polished tool, plus future WaitingRoomApp customers (consumers of the model) would likely expect the input document or page files to be automatically moved into folders representing each predicted class by the model, for example – both will require more work with help from engineers.

References

- http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
- https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf
- http://www.iajet.org/iajet_files/vol.2/no.4/Medical%20Documents%20Classification%20Based%20on%20the%20Domain%20Ontology%20MeSH.pdf
- https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/learn#text-classification
- https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/learn/text_classification_cnn.py