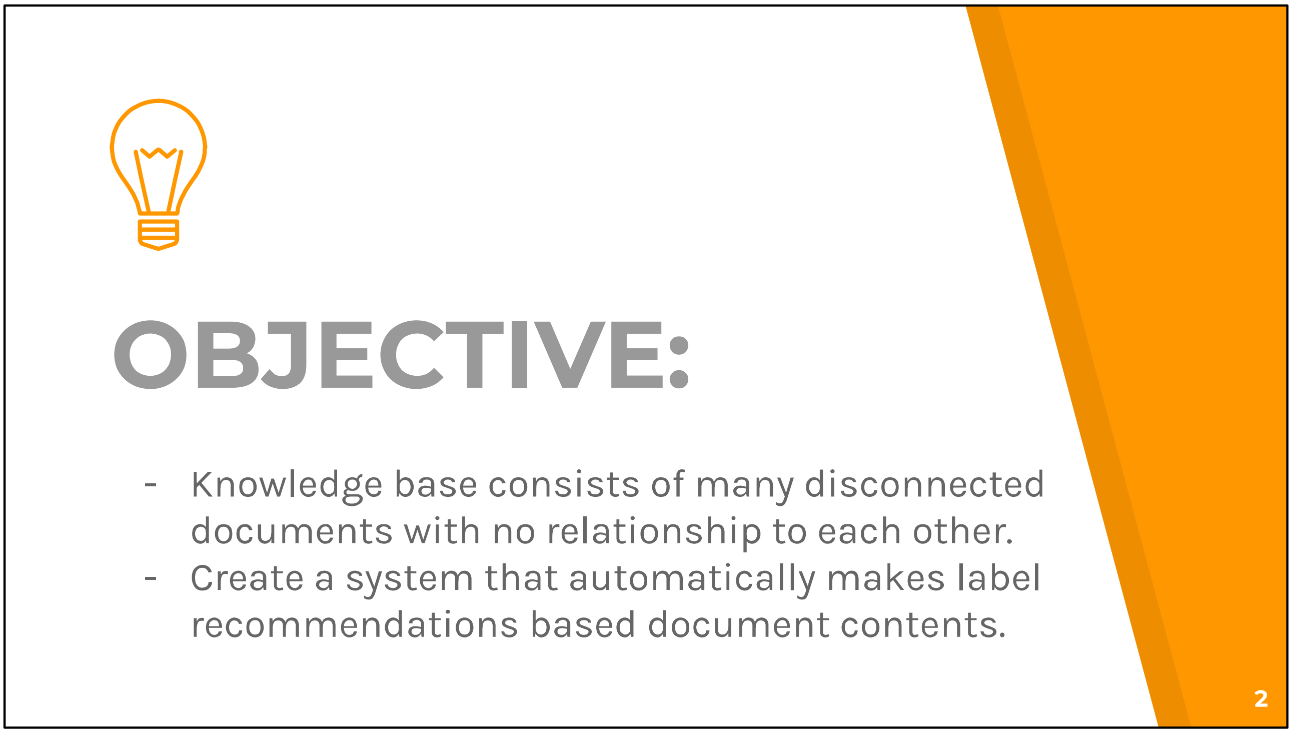
SMALLCO KNOWLEDGE BASE UPGRADE

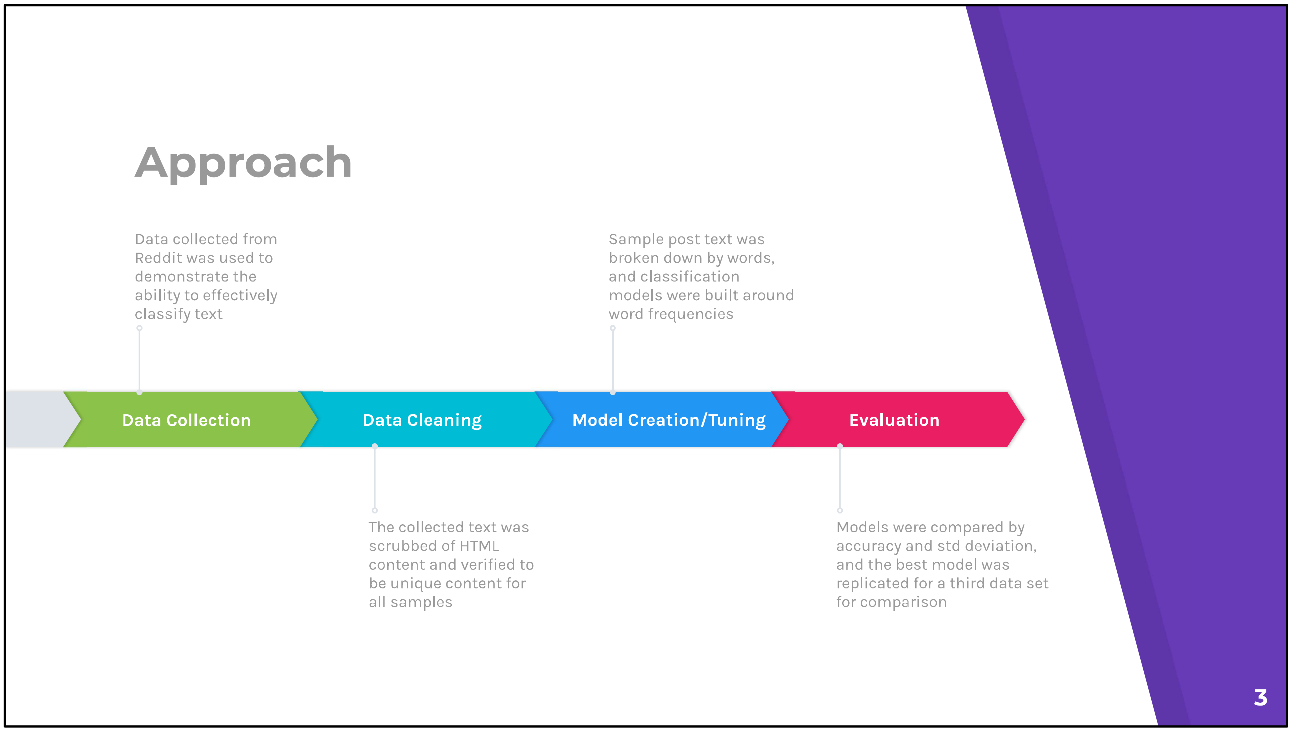
Hi there team. Thanks for joining me for this update on our Knowledge Base Upgrade project here at SMALLCO. For those who don’t know me, I’m Matt Reed, the data scientist heading this particular project.



OBJECTIVE:

- Knowledge base consists of many disconnected documents with no relationship to each other.
- Create a system that automatically makes label recommendations based document contents.

As you all know, the documents in our company knowledge base are disconnected and it is difficult to leverage prior findings and knowledge in any reasonable time. We would like to be able to reliably and automatically categorize the company’s documents for greater transfer of knowledge and lessons learned. Based on the initial work conducted here, we do believe that we have a path forward toward this end.



Approach

Data collected from Reddit was used to demonstrate the ability to effectively classify text

Sample post text was broken down by words, and classification models were built around word frequencies

Data Collection

Data Cleaning

Model Creation/Tuning

Evaluation

The collected text was scrubbed of HTML content and verified to be unique content for all samples

Models were compared by accuracy and std deviation, and the best model was replicated for a third data set for comparison

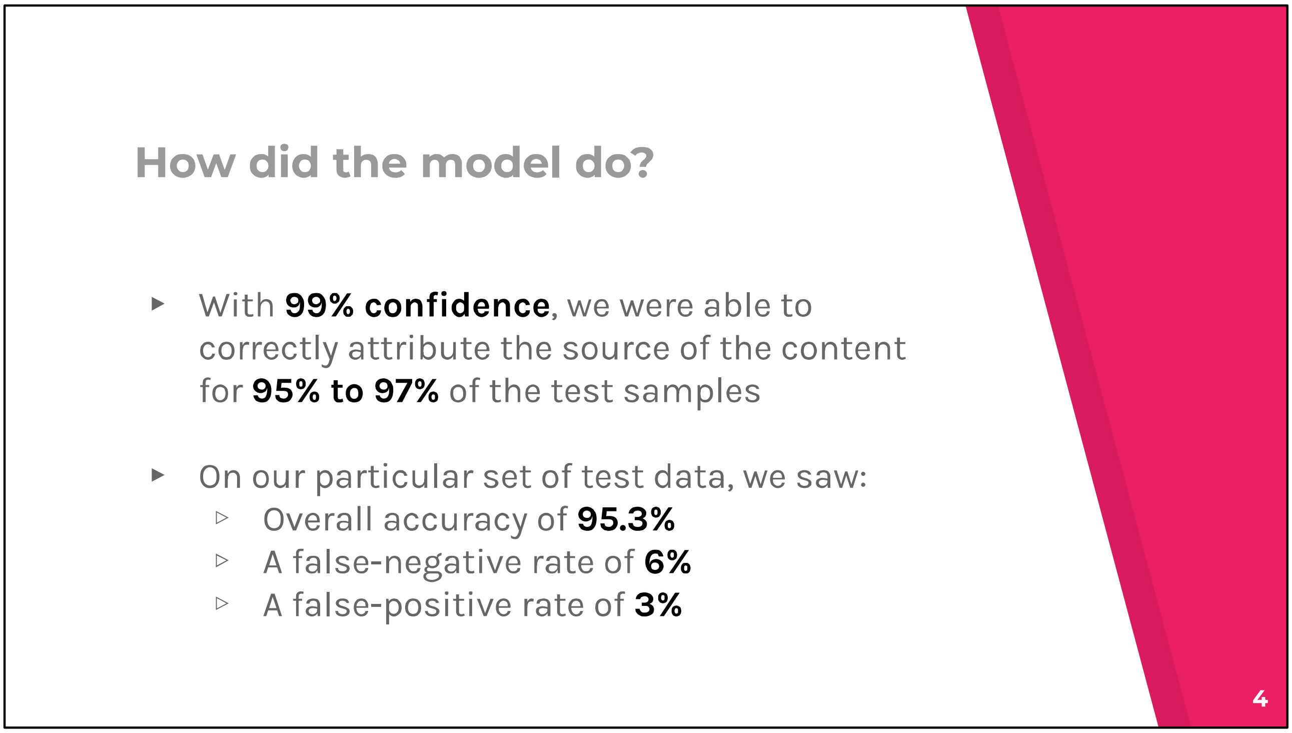
Before we get into the details, I’d like to touch on approach. We supposed that reddit forums would be a good source of surrogate data to develop and test models on, due to their pre-categorized nature.

A minimum of a thousand individual postings were collected from multiple subreddits, and used as the base data for our modeling and analysis.

The data was purged of any HTML formatting, and samples were ensured to be complete and unique.

We broke the text of the posts down into their individual words, and used the frequencies of these words inform a predictive classification model using logistic regression.

And finally, we evaluated the performance of these models based on accuracy of prediction (compared to a baseline split of 50/50), their false-positive and false-negative rates, and then applied the model towards an additional data set that was suspected to have similar content matter.



How did the model do?

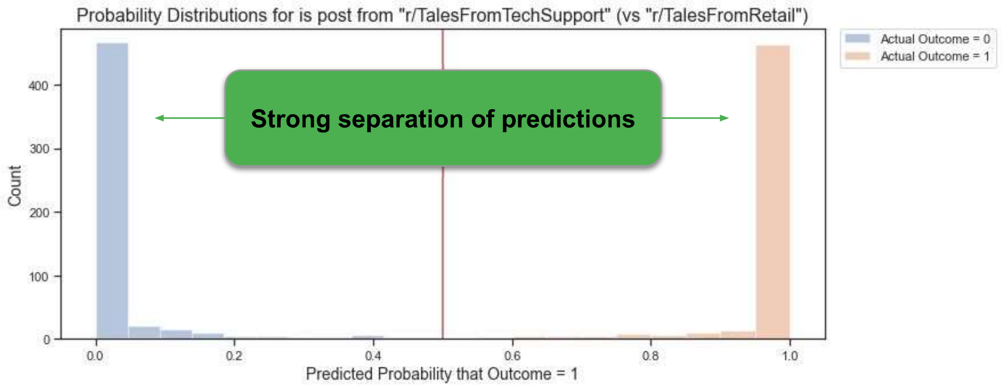
- With **99% confidence**, we were able to correctly attribute the source of the content for **95% to 97%** of the test samples
- On our particular set of test data, we saw:
  - Overall accuracy of **95.3%**
  - A false-negative rate of **6%**
  - A false-positive rate of **3%**

Our model compared two subreddits, r/TalesFromTechSupport and r/TalesFromRetail, with the hope that the similar storytelling format and long format style of the forums would produce plenty of relatively similar material for comparison.

Ultimately, our model performed very well (granted, the subreddits were not quite as dissimilar as we originally hoped). On average the model was evaluated to be over 96% accurate, with a tight standard deviation giving us a range of 95% to 97% accuracy with 99% confidence.

Our original test data set produced a false negative rate of 6%, and a false positive rate of just 3%

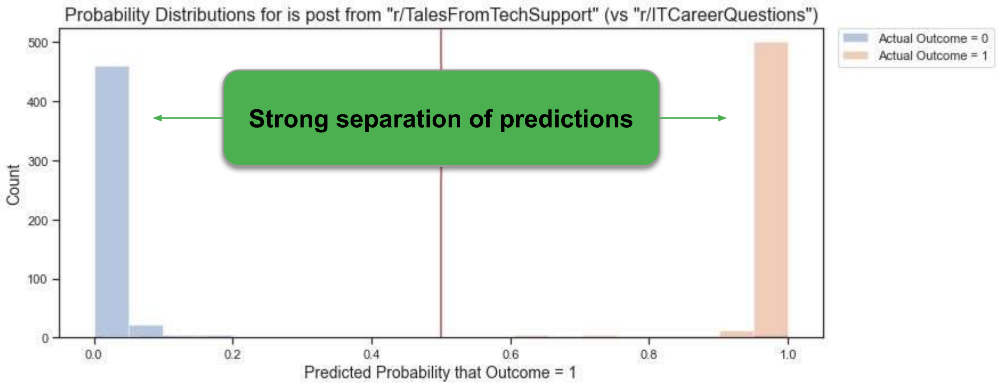
What does this look like?



Strong separation of predictions

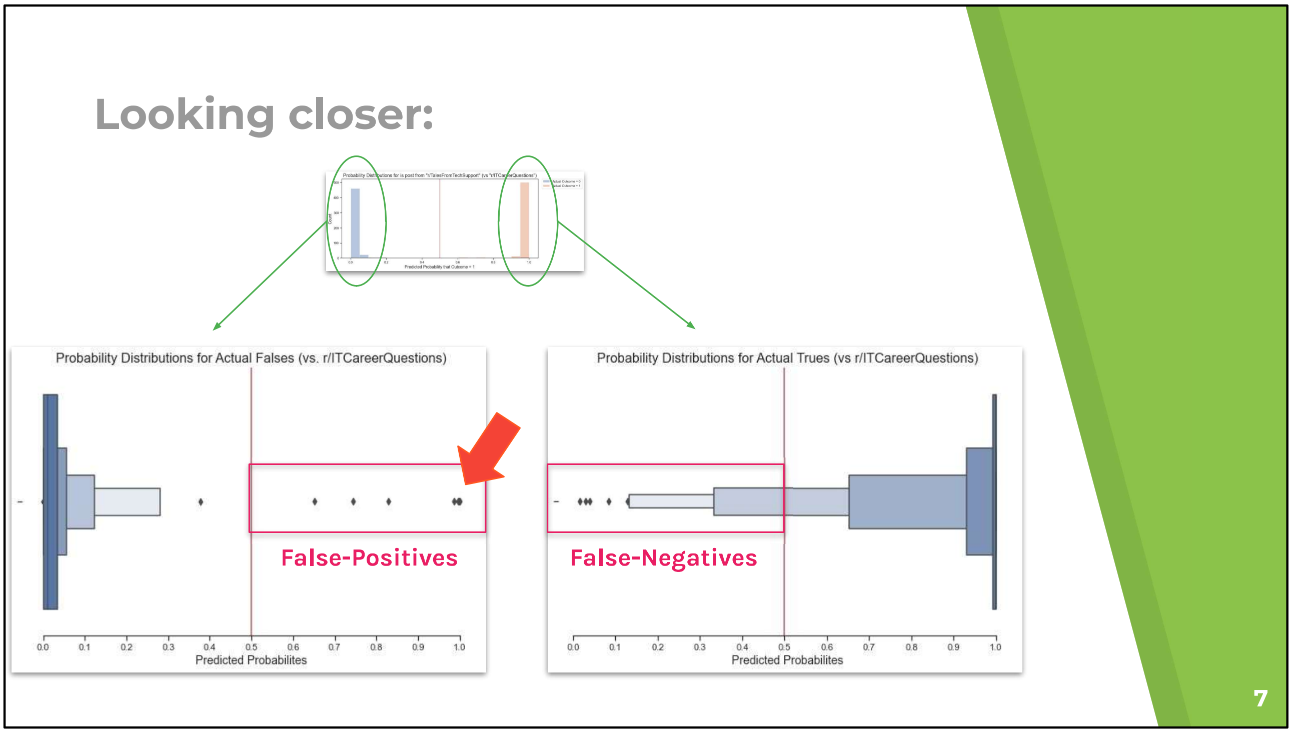
Here we can see how the model separated out predictions. Anything to the right of the center red line was predicted to be positive, while anything to the left was predicted as negative. The model had very few predictions in the middle range, and strongly pushed the probabilities to the extremes for the bulk of the data.

Does it work with other data?



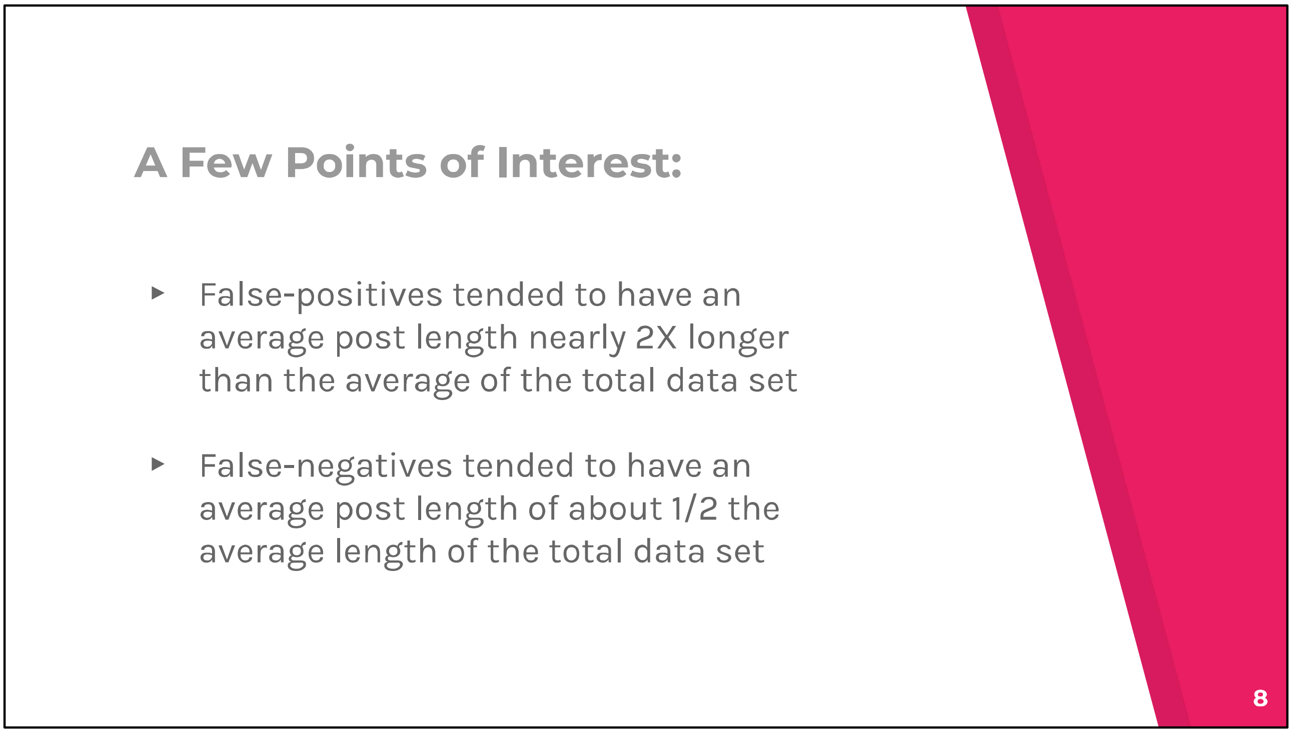
Strong separation of predictions

We wanted to see how the model we created works with another forum, particularly one we suspected would utilize similar terms (in this case ITCareerQuestions). Surprisingly, the model performed even stronger, at 97% accuracy, while having a slightly larger standard deviation giving us 99% confidence that the model is worst case 95% accurate. Unfortunately, it appears that this content also not was similar enough to actually stress test our model, but the results were encouraging nonetheless.



Looking closer:

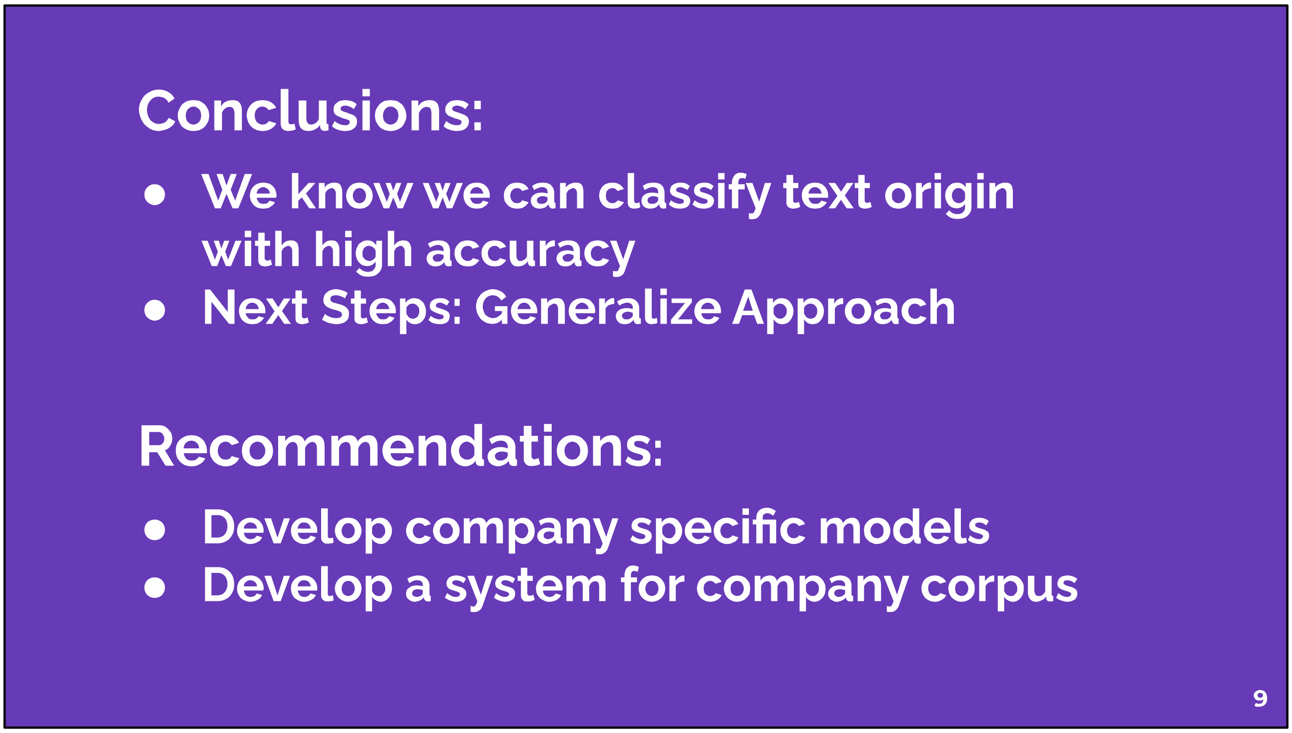
Looking a little closer, we can see that the false positives are more rare, but also have a number of cases with surprisingly high prediction probabilities. Upon investigation, we found that the content of these posts actually departed from the typical content of “ITCareerQuestions”, and would likely have been a better fit if posted in “TalesFromTechSupport”, providing strong support for the idea that our model can help to identify hidden context and smartly suggest labels.



A Few Points of Interest:

- False-positives tended to have an average post length nearly 2X longer than the average of the total data set
- False-negatives tended to have an average post length of about 1/2 the average length of the total data set

Trying to better understand our model and where its limitations may be, we zoomed in on our false positives and negatives to try and see if there were any common themes. One thing that stood out was that average word count in these cases departed significantly from the complete data set, with false positives averaging nearly 2X longer, and false negatives being about half the length on average. We'd like to look a little further into this in follow on work to see if content length does in fact skew probabilities, and whether there is an optimal length of content for analysis.



Conclusions:

- We know we can classify text origin with high accuracy
- Next Steps: Generalize Approach

Recommendations:

- Develop company specific models
- Develop a system for company corpus

In conclusion, we have demonstrated the ability to effectively categorize written content relative to established target types. We recognize that while these results are encouraging, they are limited in the fact that they are directly comparing one category to another, and we will need to develop a method for generalizing the recommendation process so that the model is focused on words unique to the target, irrespective of the negative classes it has been trained on.

We now need to create models based on company content to capture categories of interest, and develop a system for passing the knowledgebase of the company through the system for labeling.

THANKS!

Any questions?

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Death to the Stock Photo (license)