Radius Technical Report | *Our Data is Bigger Than Yours*

Michelle Leon and Victor Chen

SUMMARY

Predict the NAICS codes of business records that contain the business name, a small description and a list of websites.

MODEL & DATA PIPELINE

Summary: We preprocessed the textual data before vectorizing it using a bag of words model, and fed it to a Naive Bayes classifier to generate predictions.

Data

We utilized several data sources, including the provided challenge set data and scraped textual data from the NAICS code website. We also scraped the text from each of the business websites, and fed that into our classifier.

Pipeline

Step 1: Text Preprocessing

Our pipeline begins with text preprocessing. We first remove punctuation/digits/stopwords, and look for special optimizations in the NAICS descriptions. Then we pass the text to word lemmatizer, which maps words like “saw” to “see,” following their conceptual meaning. Then we pass the text to a standard word stemmer, which maps words like “boiling” to “boil,” removing common word endings.

Step 2: Bag of Words (TF-IDF)

In order to vectorize the textual descriptions, we used a bag of words model with tf-idf filtering. The bag of words model ignores word ordering, and simply stores a vector of counts of words to represent a document. We chose to model the text as a bag of words because it was simple/intuitive to use, and because it greatly reduced the size of our dataset. We found that the independence assumptions present in the bag of words model did not significantly affect our performance. We also adjusted our vectorized documents with tf-idf because it allowed us to 1) give rarer words higher weights, and 2) normalize between shorter and longer documents.

Step 3: Balance Training Set

It was important to balance the training set so our model wasn’t biased. We tried 2 balancing methods 1) Increase rarer samples by duplicating, 2) Merge all text related to a NAICS code into one record for training. Both gave similar results so we went with 2.

Step 4: Naive Bayes Classifier

We used a NB classifier because 1) it’s known to perform well on textual classification tasks, and 2) trains quickly on large feature spaces.

Step 5: Tune Confidence Threshold

Since there is -2 points for any wrong industry guess we though it may be worthwhile to not guess on records that have a lower confidence. Indeed this seemed to be true at first, we were getting a boost for excluding the bottom 20% of guesses. However as the model improved the benefit of not guessing dropped. On our final model it appears to make very little difference or have a negative effect no matter what the threshold is.

Step 6: Cross Validate Model and Predict

KEY CHALLENGES

No ground truth data - Not provided with any ground truth data and data was difficult to manually label in some cases.

Unusual scoring system - Business case puts *emphasis on accuracy, better to not guess records we aren’t sure about.*

Data Wrangling - NAICS descriptions contain text about what the code is not, (ie. except veterinary services) need to add more data to the records, records may have mismatched title/description.

Hierarchical label system - Must decide how to target labels, how do we include information about more vs. less specific codes.

Difficult to differentiate codes- For any product a business can make it, wholesale it, retail it, provide a service for it, consult about it, repair it, rent it. All these will contain similar words about the product but are different industries according to NAICS codes.

CONCLUSION

On our test set of 1000 manually labeled examples, we averaged around 2,500 points. We also correctly generated the three digit industry prefix for over 70% of our examples. Our main mode of failure occurred when certain businesses contained several key buzzwords from a different industry, for example, a company “Drain Medic Rx Plumbing” was misclassified as 6213 (“health practitioner”), as opposed to 238220 (“plumbing services”). We suggest further improvements in the following section.

NEXT STEPS

Given our research, we recommend future work in these areas:

- ✓ Clustering methods, k-means, label-propagation
- ✓ Better identification of important bigrams/trigrams
- ✓ Obtain more training data, through manual labeling, clustering, or web scraping
- ✓ Model to predict which predictions are wrong, thresholding in prediction withholding
- ✓ Productionize Model, parallelization in prediction