Radius Collider Submission - The Approximators

Myles Scolnick, Shane Barratt and Alexander Danilychev Jr

I. INTRODUCTION

The North American Industry Classification System (NAICS) is the standard used by Federal statistical agencies in classifying business establishments

The goal of this project was to determine the best North American Industry Classification System (NAICS) code, for a business based on its name, address, description, and website. For example, a Safeway would be classified as the code 445110 (Supermarkets and Other Grocery, except Grocery). To solve this problem, we incorporated several techniques from a variety of fields: text cleaning, data augmentation, tf-idf, wordnet, neural word embeddings and cross-validation. A pure supervised learner would be tough to model/train as there are less training data points than classes, so we went with what one could call a 'topic modeling' and 'rule based' approach.

II. DATA

To increase the amount of data on businesses and NAICS codes, we aggregated some data from public APIs including Google Places business type, NAICS code titles/descriptions, and NAICS code freqeuncies. The respective scripts are listed in List 1 in the appendix.

We also performed transformed the data: stripped stop words, lemmatized, lowercased, tokenized. We did this to normalize the text for comparisons. We also enhanced the richness of the text by adding synonyms obtained through the WordNet model [3].

III. WEB APP

A. Assisted Hand Classifier

To ease the painful process of hand-classification, we built a web application in Flask/Python that displays information about the business to be classified and a list of NAICS codes sorted by a preliminary algorithm's scoring function. This allowed more interactive classification - that is automatically written to a file. This greatly sped up hand-classifying 1,000 businesses. Refer to Figure 1 of the appendix.

B. Database View and Code Comparison

To further understand the structure of the data and the quality of our classifier, we also built an additional endpoint to read the hand-classification and algorithm-classification databases. It displays a table of businesses with our hand-classification and algorithm-classification so that we could discern what we got it wrong and why. Refer to Figure 2 of the Appendix.

IV. THE ALGORITHM

Let the set of businesses be denoted by $\{b_1, b_2, \ldots, b_n\}$ and the set of NAICS codes by $\{c_1, c_2, \ldots, c_m\}$. The algorithm attempts to construct a similarity matrix $\mathbf{S}$ of dimension nxm where $\mathbf{S}_{i,j}$ denotes the similarity between business i and NAICS code j. The classification for business (row) i is $\arg\max_{1 \leq j \leq m} \mathbf{S}_{i,j}$, the most similar NAICS code to that business. To construct this matrix, we first construct 9 similarity matrices, $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_9\}$ with the same dimension. There are 4 ways to pair the business (B) and NAICS (N) $\rightarrow$ {(B.name, B.description) X (N.name, N.description)}. Therefore, the two similarity methods, *tf-idf* cosine similarity [2] and *word2vec* cosine similarity [4], with these 4 text combinations as well as a similarity matrix of 2-digit industry priors (independent of the business) constitute the 9 similarity matrices. The final similarity matrix is a convex combination of the preliminaries:

$$\mathbf{S} = \sum_{i=1}^{9} w_i \mathbf{S}_i,$$

where $\sum_{i=1}^{9} w_i = 1$. The weights were selected through random hyper-parameter search [1].

The codebase is in python, and used some packages including gensim (word2vec), nltk (nlp) and scikit-learn (tf-idf). Github was also used for version control.

V. CONCLUSION

A. Results

After training, our algorithmic classifier scores 30% against our hand-classified 1000 businesses.

B. Confidence in our Predictions

- We are not particularly confident in the predictions our algorithm made, as it has lots of false-positives.

C. Limitations / Unexplored Ideas

- The algorithm did not take into account the hierarchical structure of the NAICS codes. We believe this could help the score by determing the marginal expected value for each additional digit.
- The algorithm did not take into account the hierarchical structure of the NAICS codes. We believe this could help the score by determing the marginal expected value for each additional digit.
- Collecting data was pretty limited, but if possible to gather the approximate amount of revenue and/or costs coming from a business, it would be helpful in narrowing down the correct NAICS industry.

D. Challenges

- The data did not come perfectly formatted and sometimes the title, description, website did not actually match and was a mix of 2 businesses.
- Some of the businesses had names and descriptions in other languages.
- Some businesses have multiple obvious classifications making it difficult to assign a single NAICS code (*e.g.* a bakery which also is a sit-down restaurant).

REFERENCES

[1] Bergstra, James, and Yoshua Bengio. "Random Search for Hyper-parameter Optimization." *The Journal of Machine Learning Research* 13.1 (2012): 281-305.

[2] Brown, Peter F., et al. "A Statistical Approach to Machine Translation." *Computational Linguistics* 16.2 (1990): 79-85.

[3] Miller, George A., et al. "Introduction to Wordnet: An On-line Lexical Database*." *International Journal of Lexicography* 3.4 (1990): 235-244.

[4] Mikolov, Tomas, et al. "Distributed Representations of Words and Phrases and their Compositionality." *Advances in Neural Information Processing Systems*. 2013.