Project Milestone 3

Manjinder Sandhu, Dagim Bantikassegn, Tyler Dabbs, Alex Brooks

In our last milestone, we talked about an NER model and using the eCFR text data set to create our model. Our group has had multiple video message discussions where we decided to go in a bit of a different direction. We had previously worked on Homework 2: Text Classification together. After working on that homework assignment, we realized that we enjoyed text classification more and also, felt we could create a more accurate model. With milestone 2 we were able to see how the data was separated and were able to see the different chapters and subchapters. ChatGPT was also used to help us with this project milestone.

Earlier we discussed how we originally decided to create an NER model. We had multiple discussions about it, however, we ultimately decided that text classification would be the better fit. This is because of how diverse the dataset is. The diversity of the dataset made us feel like we would have way too many different topics. We looked through the chapter list we created in project milestone 2, and felt that maybe we should reconsider our options. After looking at the suggested use cases, offered by the professor, we decided to switch directions toward text classification.

For Text Classification, we want to be able to create a train dataset and a test dataset. Because of our work looking at the different chapters and subchapters, we realized that we needed a way to spread the dataset out. Of course, you cannot manually do this as you want it to be purely random. We used a seed, however, in case another person wants to replicate the code, so it will be the same spread. Sci-Kit Learn offers a Train-Test split option that allows us to randomly select 200 lines in the JSONL file to be our evaluation dataset and the remaining 4,465 lines will be our train dataset.

Once we had created the train and test set, we needed to create our categories. To create our categories, we used the ChatGPT Prompt shown in the Appendix. If you look, you'll see we asked for a description of each of the categories, a description you would give a high schooler (this is for people with less banking experience), an example of each category, and how each category affects the bank as a whole. We took the 5 categories that ChatGPT gave us, and used Ngrok for all of us to be able to classify all of the 200 text lines.

A blocker we encountered was when we did homework 2, we were only able to have 1 person do text classification at 1 time. This was fine during homework 2, but the text data in each was very rich and long, and we also had multiple categories. This caused us to have to have ngrok open for over 9 hours. It took us on average over 2 hours and 15 minutes for each of us to finish our labeling. To be able to find a whole day in which we could be able to accomplish this together was a huge challenge, luckily our schedules aligned one Saturday, to allow it to work. Once this was completed, we went through each of the 200 to create our "gold standard" evaluation dataset. Using multiple different people classifying data, you can have indecisiveness for how data should be classified. Because it can be subjective, we always tried to go with majority, however, if we had a tie we would send the classifications that were tied and ask ChatGPT to classify the text with those categories. This allowed us to have an unbiased "tie-breaker" creating the data set.

Using prodigy we first trained a model using 200 lines of text from the train data. We classified these and then tested this model with the evaluation set. The results are found on the second page of the appendix. As you can see we have a score of 0.67, which is very good for a text classification with five different categories. We then created a model using the 200 lines with SpaCy train and the base model as “en_core_web_lg”. The SpaCy model statistics can be seen on the third page of the appendix. We can see a slight increase in our f score. Next we annotated the remaining dataset by using Weakly label. When we trained the weakly label using prodigy we got a score of 0.66 and for SpaCy train we got a f score of 0.71. The f score slightly remained the same in training in both spaCy and prodigy train when we did the 200 annotations and 4465 annotations. We believe that the first 200 annotations were basic and more straightforward while in the 4465 annotations were more diverse which had more challenging data that’s why it remained the same overall. The terminal commands are on the .txt file and the python code is on github. The link can be found in the appendix

To look at predictions from our best model, we collected five different comments made about proposals that might be introduced to our document to determine which of the five labels it most relates to. Using those new examples, we created code using the help of the OpenAI assistant and saw that one of them was “Risk Management” and four of them were “Reporting and Compliance”. Manually reading these comments though, one of them does seem to fit more with “Consumer Protection” than “Reporting and Compliance”, therefore there is room for improvement

In regards to next steps, we had a couple of potential ideas. One of our ideas was to try including some custom stopwords so that certain tokens don’t influence the models in an unfavorable way. This is an idea that came about mainly due to the nature of these specific texts. There are multiple words and symbols used that are prevalent in several of the texts such as “chapter”, roman numerals, and there was also this symbol “§” (stands for section, used to reference a previous section) which was utilized heavily throughout the texts. The worry with some of these words and symbols is that when we were going through classifying the texts individually, some of the choices we made could cause the model to look at the training data and say, ok the word “chapter” means CorporateGovernance. We don’t want our models making definitive conclusions just based on these single tokens.

Another potential idea for the future would be using engrams to separate the text into phrases or sequences of words. This could theoretically help the model classify the texts a bit easier since for example, three word tokens can give the model more information to work with when going through reading each text. We could potentially try a few different lengths for the engrams to see what combinations work best.

Our timeline for the next 2 weeks is to refine the model and work on these potential ideas we have to strengthen the model. Once we receive feedback from the professor, we plan to use their feedback for issues that we may not be seeing in our model. We are wanting to reserve 2 weeks just in case there is a lot of feedback that we need to consider. We will be heavily communicating with one another through google spaces to allow us to accomplish these goals.

Appendix

Github:
https://github.com/ManjinderUNCC/project3-DSBA6188

ChatGPT Prompt:
https://chat.openai.com/share/35969a1f-4fa0-41d2-8a68-08492afb33e2

Prodigy Train: 200

```
========================= Generating Prodigy config =========================
ℹ Auto-generating config with spaCy
✔ Generated training config

============================ Initializing pipeline ============================
[2024-03-31 14:37:28,675] [INFO] Set up nlp object from config
Components: textcat_multilabel
Merging training and evaluation data for 1 components
  - [textcat_multilabel] Training: 199 | Evaluation: 200 (from datasets)
Training: 199 | Evaluation: 200
Labels: textcat_multilabel (5)
[2024-03-31 14:37:28,807] [INFO] Pipeline: ['textcat_multilabel']
[2024-03-31 14:37:28,810] [INFO] Created vocabulary
[2024-03-31 14:37:28,811] [INFO] Finished initializing nlp object
[2024-03-31 14:37:29,879] [INFO] Initialized pipeline components: ['textcat_multilabel']
✔ Initialized pipeline

============================== Training pipeline ==============================
Components: textcat_multilabel
Merging training and evaluation data for 1 components
  - [textcat_multilabel] Training: 199 | Evaluation: 200 (from datasets)
Training: 199 | Evaluation: 200
Labels: textcat_multilabel (5)
ℹ Pipeline: ['textcat_multilabel']
ℹ Initial learn rate: 0.001
E    #       LOSS TEXTC...  CATS_SCORE  SCORE
---  ------  -------------  ----------  ------
  0       0           0.25       46.65    0.47
  1     200          29.88       53.74    0.54
  2     400          17.91       57.57    0.58
  3     600           9.81       63.93    0.64
  4     800           5.28       65.00    0.65
  5    1000           3.87       66.95    0.67
  6    1200           2.98       65.20    0.65
  7    1400           1.06       67.70    0.68
  8    1600           1.62       68.16    0.68
  9    1800           1.56       67.14    0.67
 10    2000           1.10       67.08    0.67
 12    2200           1.09       66.96    0.67
 13    2400           1.47       66.95    0.67
 14    2600           1.06       66.95    0.67
 15    2800           1.06       67.18    0.67
 16    3000           1.23       67.31    0.67
 18    3200           1.64       67.21    0.67
✔ Saved pipeline to output directory
```

Spacy Train: 200

```
============================ Initializing pipeline ============================
✔ Initialized pipeline

============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner', 'textcat_multilabel']
ℹ Frozen components: ['tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS TEXTC...  CATS_SCORE  SPEED   SCORE
---  ------  ------------  -------------  ----------  ------  ------
  0       0          0.05           0.40       50.83  6853.12    0.51
  5    1000          3.80         104.20       68.15  7236.50    0.68
 10    2000          6.95          23.44       69.55  7260.22    0.70
 16    3000          1.22           2.73       70.69  7173.95    0.71
 23    4000          0.00           0.04       69.89  7131.99    0.70
 32    5000          0.00           0.02       70.08  7153.21    0.70
 42    6000          0.00           0.01       70.04  7112.60    0.70
 53    7000          8.55          17.11       68.78  7115.19    0.69
 63    8000          2.41           2.77       69.47  7177.40    0.69
```

Prodigy Train: 4465

```
========================= Generating Prodigy config =========================
ℹ Auto-generating config with spaCy
✔ Generated training config

============================ Initializing pipeline ============================
[2024-03-31 15:45:24,080] [INFO] Set up nlp object from config
Components: textcat_multilabel
Merging training and evaluation data for 1 components
  - [textcat_multilabel] Training: 4465 | Evaluation: 200 (from datasets)
Training: 4465 | Evaluation: 200
Labels: textcat_multilabel (5)
[2024-03-31 15:45:24,810] [INFO] Pipeline: ['textcat_multilabel']
[2024-03-31 15:45:24,814] [INFO] Created vocabulary
[2024-03-31 15:45:24,814] [INFO] Finished initializing nlp object
[2024-03-31 15:45:47,609] [INFO] Initialized pipeline components: ['textcat_multilabel']
✔ Initialized pipeline

============================= Training pipeline =============================
Components: textcat_multilabel
Merging training and evaluation data for 1 components
  - [textcat_multilabel] Training: 4465 | Evaluation: 200 (from datasets)
Training: 4465 | Evaluation: 200
Labels: textcat_multilabel (5)
ℹ Pipeline: ['textcat_multilabel']
ℹ Initial learn rate: 0.001
```

E	#	LOSS TEXTC...	CATS_SCORE	SCORE
0	0	0.25	46.74	0.47
0	200	19.69	51.16	0.51
0	400	19.75	53.63	0.54
0	600	17.71	53.21	0.53
0	800	19.14	53.54	0.54
0	1000	16.71	53.38	0.53
0	1200	17.21	53.41	0.53
0	1400	16.78	54.40	0.54
0	1600	14.85	53.70	0.54
0	1800	15.98	54.37	0.54
0	2000	14.77	53.83	0.54
0	2200	15.34	55.63	0.56
0	2400	16.28	53.35	0.53
0	2600	13.70	55.07	0.55
0	2800	13.08	55.60	0.56
0	3000	16.99	55.79	0.56
0	3200	14.84	55.29	0.55
0	3400	13.99	54.94	0.55
0	3600	14.64	56.58	0.57
0	3800	12.97	57.23	0.57
1	4000	11.27	58.31	0.58
1	4200	12.18	58.56	0.59
1	4400	9.27	59.94	0.60
1	4600	9.02	59.97	0.60
1	4800	9.11	60.49	0.60
1	5000	11.34	63.29	0.63
1	5200	10.81	60.84	0.61
1	5400	10.51	61.48	0.61
1	5600	8.01	62.93	0.63
1	5800	10.90	62.76	0.63
1	6000	8.72	63.56	0.64
1	6200	9.58	62.53	0.63
1	6400	8.66	62.37	0.62
2	6600	6.52	64.19	0.64
2	6800	6.99	63.73	0.64
2	7000	5.70	64.57	0.65
2	7200	6.70	62.57	0.63
2	7400	5.57	64.02	0.64
2	7600	6.28	64.12	0.64
2	7800	5.74	66.26	0.66
2	8000	7.23	63.25	0.63
2	8200	7.33	65.08	0.65
2	8400	5.59	66.66	0.67
2	8600	8.49	63.45	0.63
3	8800	6.80	66.00	0.66
3	9000	4.85	64.79	0.65
3	9200	5.16	65.06	0.65
3	9400	4.79	64.96	0.65
3	9600	3.95	66.44	0.66
3	9800	4.94	65.29	0.65
3	10000	4.75	64.80	0.65

Spacy Train: 4465

```
============================ Initializing pipeline ============================
✔ Initialized pipeline

============================== Training pipeline ==============================
ℹ Pipeline: ['tok2vec', 'tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner', 'textcat_multilabel']
ℹ Frozen components: ['tagger', 'parser', 'attribute_ruler',
'lemmatizer', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS TEXTC...  CATS_SCORE  SPEED   SCORE
---  ------  ------------  -------------  ----------  ------  ------
  0       0          0.04           0.37       58.23  6816.17    0.58
  0    1000          1.37          97.28       69.04  6974.42    0.69
  0    2000          1.57          77.44       71.07  7015.26    0.71
  0    3000          0.88          69.37       67.89  7029.25    0.68
  1    4000          1.24          59.19       67.73  7020.66    0.68
  1    5000          1.68          46.44       66.59  6970.28    0.67
  1    6000          2.41          42.51       68.16  7098.61    0.68
  2    7000          2.65          34.48       67.86  7100.93    0.68
```

New Inputs for Classification:

```
Text: Banks that are at risk of failing selling bonds? Absolutely not! No way! The idea of where this money needs to come from should've been a thought that was had before these

ReportingAndCompliance: 0.3665
RiskManagement: 0.0330
ConsumerProtection: 0.0310
CorporateGovernance: 0.0423
CapitalRequirements: 0.0245

Text: The Wisconsin Bankers Association (aka the WBA) is the largest financial trade association in Wisconsin, representing over 200 state and nationally chartered banks, savings

ReportingAndCompliance: 0.6879
RiskManagement: 0.0000
ConsumerProtection: 0.0048
CorporateGovernance: 0.0000
CapitalRequirements: 0.0000

Text: How about you crooks focus on the billions being laundered by banks in plain fucking sight instead of intruding in our lives more. Disgusting. Aweful.

ReportingAndCompliance: 0.4072
RiskManagement: 0.2440
ConsumerProtection: 0.3574
CorporateGovernance: 0.3809
CapitalRequirements: 0.2414

Text: If adopted, this proposal [R-1726], would prove to be an invasion of privacy. In terms of digital assets, crypto exchanges are not held accountable in the same way that oth

ReportingAndCompliance: 0.4365
RiskManagement: 0.5856
ConsumerProtection: 0.3847
CorporateGovernance: 0.1818
CapitalRequirements: 0.1904

Text: Amendments to 20402(d)(2) and 204.2(e)(2) and (4) make a savings account without transfer or withdrawal limits transaction accounts. Can a depository institution  avoid hav

ReportingAndCompliance: 0.2221
RiskManagement: 0.0007
ConsumerProtection: 0.0513
CorporateGovernance: 0.0031
CapitalRequirements: 0.0000
```