DDSI Internship Experience

7/22/20

Will Hibbard

Background/ Overview

- The UMLS metathesaurus is a database of medical vocabularies and standards
- Goal: to semantically group the atoms in the UMLS database based on lexical and contextual information to allow for better synonymy prediction
- Why: sorting the terms into their semantic groups allows for better synonymy prediction with the atoms, but previous lexical only methods caused false synonymy between terms
  - Ex. Splint (shin) and splint (medical device)
- How: Using deep learning, we embed atoms into vectors and train a neural network on those until it can accurately predict the semantic groups of those atoms. Then, the model is tested with a dataset it’s never seen before

Learning Neural Networks

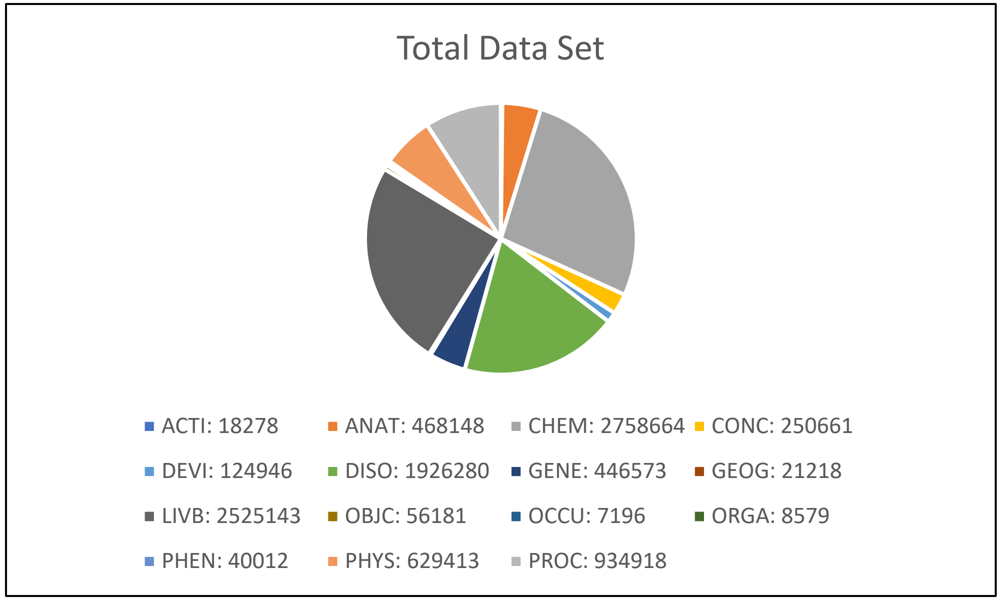
- I started by reading up on Neural Networks and going over example code with Yuqing
- The code I based my neural network off of was from a programming blog that made a neural network to classify wines by type
- Admittedly, I spent the first few weeks getting access to TOAD and Biowulf
- My network ended up being a 5-layer classification model with SAP BERT embedding
- SAP BERT was chosen because it maintained the sentence and paragraph structure of the atoms it embedded, allowing for contextual info like word placement and order to be considered

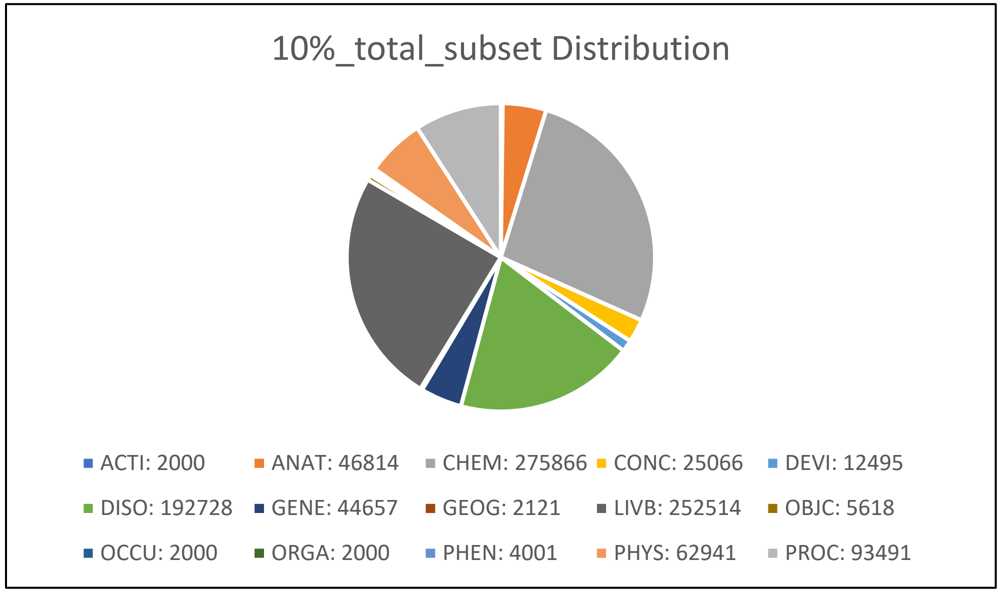
The Dataset

- Gathered with TOAD on a virtual machine
- The desired dataset could only have atoms that were in English and came from active source vocabularies
- The information in the dataset was AUI’s, the string, and TUIs
  - AUIs were used as unique identifiers
  - The string was the data we wanted
  - TUIs were used to map the atoms to their correct groups so the model could gauge its performance
- The entire dataset was 10,000,000+ atoms and was too big to run normally, so two subsets were made for code training and testing

The Subsets

- The subsets used to train the model were ~1,000,000+ atoms, and were quicker and more manageable to run
- The subsets were assembled with stratified sampling to ensure that the model had practice sorting every one of the semantic groups
- 10 % subset: This subset was assembled with 10% of each SG to give the model something small to run that was proportionally accurate to the whole set
- Balanced subset: This subset was made of 7000 atoms from each SG to see how the model performed without bias weights towards larger SGs





Experimentation

- Experiments for the classification problem were done by running the datasets with different values for Epoch, Batch Size, Learning Rate, Hidden Layers, and nodes per Hidden Layer
    - Default: Epoch = 30, Batch Size = 512, Learning Rate = 0.0007, # of hidden layers = 3, nodes per layer = 64, 128, 512, etc…
- The parameters were adjusted individually to prevent overfitting the model
- After the optimal hyperparameters were collected, they were tested together on the datasets

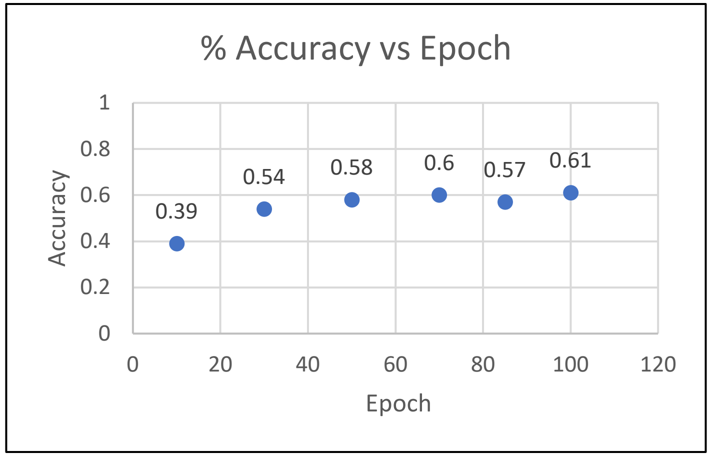
Output

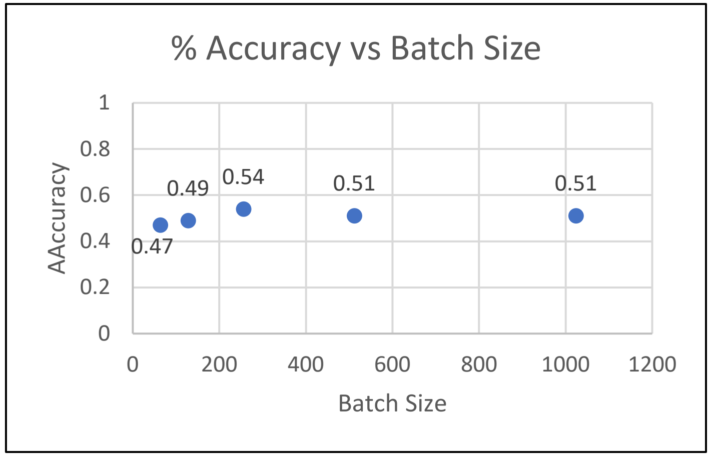
- The results of each run were broken down into precision, recall, F-1, average, macro average, and weighted average
  - Precision – the model's accuracy in classifying an atom to a group
  - Recall – was the positive classification correct
  - F-1 – a score that says the weighted average of precision and recall
  - Accuracy – how the model did across all classes
  - Macro average – the unweighted average of each column
  - Weighted average – the average that accounts for the class support
- The supports don't reflect the whole subset size as the code divides the subset into training, validation, and testing sets

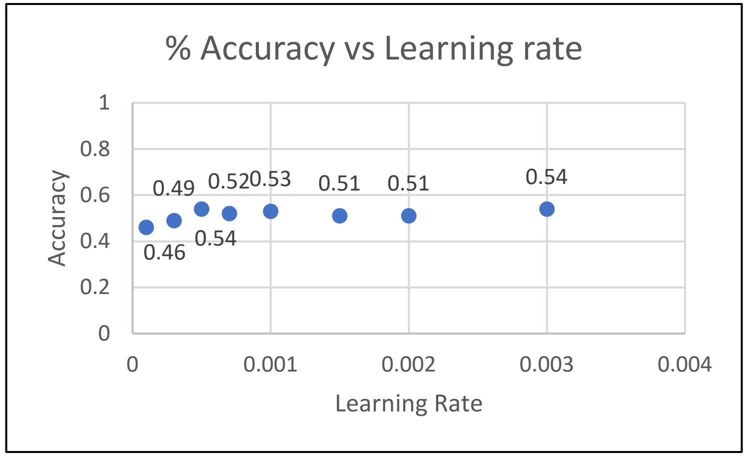
Epoch=30, Batch size=512				
	precision	recall	f1-score	support
ACTI	0.19	0.52	0.27	400
ANAT	0.34	0.68	0.45	9363
CHEM	0.89	0.50	0.64	53736
CONC	0.08	0.50	0.14	5013
DEVI	0.22	0.75	0.34	2499
DISO	0.85	0.47	0.61	38546
GENE	0.43	0.88	0.58	8932
GEOG	0.04	0.24	0.06	424
LIVB	0.88	0.46	0.60	50503
OBJC	0.07	0.39	0.11	1124
OCCU	0.07	0.24	0.11	400
ORGA	0.07	0.29	0.11	400
PHEN	0.12	0.33	0.18	800
PHYS	0.73	0.85	0.79	12588
PROC	0.59	0.61	0.60	18699
accuracy			0.54	203427
macro avg acc	0.37	0.51	0.37	203427
weighted avg acc	0.76	0.54	0.59	203427
set = 10% total subset	Time=~20 min			

Results

- Some incremental improvement in accuracy across optimal hyperparameters, but the overall accuracy was not practically applicable
- I was unable to get the whole dataset running until this week, so the results depict the subsets







Results pt. 2

Epoch = 30, BatchSize = 512

	precision	recall	f1-score	support
ACTI	0.38	0.34	0.36	1400
ANAT	0.54	0.59	0.56	1400
CHEM	0.49	0.56	0.52	1400
CONC	0.35	0.31	0.33	1400
DEVI	0.60	0.58	0.59	1400
DISO	0.52	0.45	0.48	1400
GENE	0.72	0.71	0.71	1400
GEOG	0.34	0.35	0.35	1400
LIVB	0.62	0.58	0.60	1400
OBJC	0.36	0.39	0.37	1400
OCCU	0.38	0.35	0.37	1400
ORGA	0.32	0.36	0.34	1323
PHEN	0.36	0.37	0.37	1400
PHYS	0.74	0.84	0.79	1400
PROC	0.41	0.36	0.38	1400
accuracy			0.48	20923
macro avg	0.48	0.48	0.48	20923
weighted avg	0.48	0.48	0.48	20923

set = balanced data set

Epoch = 100, BatchSize = 256

	precision	recall	f1-score	support
ACTI	0.33	0.37	0.35	1400
ANAT	0.62	0.54	0.58	1400
CHEM	0.47	0.56	0.51	1400
CONC	0.34	0.31	0.33	1400
DEVI	0.62	0.58	0.60	1400
DISO	0.44	0.50	0.47	1400
GENE	0.72	0.71	0.72	1400
GEOG	0.35	0.35	0.35	1400
LIVB	0.58	0.61	0.60	1400
OBJC	0.37	0.37	0.37	1400
OCCU	0.42	0.36	0.38	1400
ORGA	0.37	0.32	0.34	1323
PHEN	0.38	0.39	0.38	1400
PHYS	0.74	0.82	0.78	1400
PROC	0.40	0.38	0.39	1400
accuracy			0.48	20923
macro avg	0.48	0.48	0.48	20923
weighted avg	0.48	0.48	0.48	20923

set = balanced data set

Epoch = 70, BatchSize = 1024

	precision	recall	f1-score	support
ACTI	0.34	0.33	0.34	1400
ANAT	0.50	0.62	0.56	1400
CHEM	0.49	0.55	0.52	1400
CONC	0.38	0.28	0.32	1400
DEVI	0.55	0.59	0.57	1400
DISO	0.48	0.49	0.48	1400
GENE	0.69	0.71	0.70	1400
GEOG	0.36	0.31	0.33	1400
LIVB	0.59	0.59	0.59	1400
OBJC	0.35	0.36	0.36	1400
OCCU	0.40	0.35	0.38	1400
ORGA	0.37	0.34	0.35	1323
PHEN	0.38	0.39	0.38	1400
PHYS	0.75	0.81	0.78	1400
PROC	0.40	0.39	0.39	1400
accuracy			0.48	20923
macro avg	0.47	0.48	0.47	20923
weighted avg	0.47	0.48	0.47	20923

set = balanced data set

LR = 0.0005, Epoch=30, BatchSize=512

	precision	recall	f1-score	support
ACTI	0.35	0.35	0.35	1400
ANAT	0.60	0.54	0.57	1400
CHEM	0.52	0.54	0.53	1400
CONC	0.34	0.33	0.33	1400
DEVI	0.59	0.57	0.58	1400
DISO	0.50	0.50	0.50	1400
GENE	0.72	0.73	0.73	1400
GEOG	0.31	0.37	0.34	1400
LIVB	0.60	0.61	0.60	1400
OBJC	0.40	0.34	0.37	1400
OCCU	0.33	0.38	0.35	1400
ORGA	0.33	0.36	0.35	1323
PHEN	0.43	0.35	0.39	1400
PHYS	0.77	0.81	0.79	1400
PROC	0.40	0.37	0.39	1400
accuracy			0.48	20923
macro avg	0.48	0.48	0.48	20923
weighted avg	0.48	0.48	0.48	20923

set = balanced data set

Results pt. 3

Layer 2 Removed				
LR = 0.0007, Epoch=30, BatchSize = 512				
	precision	recall	f1-score	support
ACTI	0.34	0.37	0.35	1400
ANAT	0.51	0.64	0.57	1400
CHEM	0.55	0.48	0.51	1400
CONC	0.34	0.35	0.35	1400
DEVI	0.61	0.55	0.58	1400
DISO	0.50	0.48	0.49	1400
GENE	0.70	0.71	0.71	1400
GEOG	0.37	0.32	0.34	1400
LIVB	0.57	0.62	0.60	1400
OBJC	0.39	0.35	0.37	1400
OCCU	0.37	0.37	0.37	1400
ORGA	0.38	0.34	0.36	1323
PHEN	0.38	0.38	0.38	1400
PHYS	0.77	0.81	0.79	1400
PROC	0.36	0.41	0.38	1400
accuracy			0.48	20923
macro avg	0.48	0.48	0.48	20923
weighted avg	0.48	0.48	0.48	20923
set = balanced data set				

Layer 2 removed				
LR=0.0005, Epoch=100, BatchSize=256				
	precision	recall	f1-score	support
0	0.36	0.38	0.37	1400
1	0.57	0.57	0.57	1400
2	0.48	0.52	0.50	1400
3	0.35	0.32	0.34	1400
4	0.60	0.57	0.58	1400
5	0.50	0.46	0.48	1400
6	0.68	0.73	0.70	1400
7	0.37	0.33	0.35	1400
8	0.57	0.62	0.59	1400
9	0.37	0.37	0.37	1400
10	0.34	0.36	0.35	1400
11	0.36	0.36	0.36	1323
12	0.38	0.38	0.38	1400
13	0.77	0.83	0.80	1400
14	0.41	0.35	0.38	1400
accuracy			0.48	20923
macro avg	0.47	0.48	0.47	20923
weighted avg	0.47	0.48	0.47	20923
set = balanced data set				

Future Directions

- Running the model to find the upper limits of the variables I changed for experimentation
- Create a confusion matrix of what atoms got misclassified into which classes
- Binary prediction where the model determines if atoms in the dataset belong to one class, for each class
- Fine tuning the model based on which classes the model isn't confident about assigning atoms to
- Use BioWordVec with SAP BERT to combine strings with SAB

Epoch = 300, SAB & STR w/ S-B & BWV				
	precision	recall	f1-score	support
0	0.46	0.62	0.53	400
1	0.70	0.85	0.77	9363
2	0.94	0.84	0.89	53736
3	0.24	0.46	0.32	5013
4	0.60	0.77	0.67	2499
5	0.81	0.69	0.74	38546
6	0.86	0.97	0.91	8932
7	0.17	0.34	0.23	424
8	0.96	0.78	0.86	50503
9	0.18	0.36	0.24	1124
10	0.16	0.30	0.20	400
11	0.17	0.28	0.21	400
12	0.21	0.36	0.27	800
13	0.93	0.97	0.95	12588
14	0.53	0.81	0.64	18699
accuracy			0.79	203427
macro avg	0.53	0.63	0.56	203427
weighted avg	0.84	0.79	0.80	203427
set = 10 % subset				

References

- Bajaj, A. (2022, March 18). Performance Metrics in Machine Learning [Complete Guide] [web log]. Retrieved June 21, 2022, from https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide#:~:text=%20Performance%20Metrics%20in%20Machine%20Learning%20%20,now%20understand%20the%20importance%20of%20performance...%20More%20.
- Brownlee, J. (2020, September 11). *Understand the impact of learning rate on neural network performance*. Machine Learning Mastery. Retrieved July 22, 2022, from https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/
- Gad, A. (2021, February 19). *Evaluating deep learning models: The confusion matrix, accuracy, precision, and recall*. KDnuggets. Retrieved July 22, 2022, from https://www.kdnuggets.com/2021/02/evaluating-deep-learning-models-confusion-matrix-accuracy-precision-recall.html
- HPC @ NIH (2022, June 6). *Locally mounting HPC system directories*. National Institutes of Health. Retrieved June 15, 2022, from https://hpc.nih.gov/docs/hpcdrive.html
- Koehrsen, W. (2018, October 2). *Neural network embeddings explained*. Medium. Retrieved May 25, 2022, from https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526
- Leung, K. (2022, June 20). *Micro, Macro & weighted averages of F1 score, clearly explained*. Medium. Retrieved July 22, 2022, from https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f
- Malik, F. (2021, March 4). *Neural networks bias and weights*. Medium. Retrieved July 22, 2022, from https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da
- Mao, Y., & Fung, K. W. (2020). Use of word and graph embedding to measure semantic relatedness between unified medical language system concepts. *Journal of the American Medical Informatics Association*, *27*(10), 1538–1546. https://doi.org/10.1093/jamia/ocaa136
- Nguyen, V., Yip, H. Y., & Bodenreider, O. (2021). Biomedical vocabulary alignment at scale in the UMLS metathesaurus. *Proceedings of the Web Conference 2021*. https://doi.org/10.1145/3442381.3450128
- Nguyen, V., Yip, H. Y., Bajaj, G., Wijesiriwardene, T., Javangula, V., Parthasarathy, S., Sheth, A., & Bodenreider, O. (2022). Context-enriched learning models for aligning biomedical vocabularies at scale in the UMLS metathesaurus. *Proceedings of the ACM Web Conference 2022*. https://doi.org/10.1145/3485447.3511946
- UMLS® Reference Manual [Internet]. Bethesda (MD): National Library of Medicine (US); 2009 Sep-. Table 1. [Concept Names and Sources (File = MRCONSO.RRF)]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK9685/table/ch03.T.concept_names_and_sources_file_mr/
- UMLS® Reference Manual [Internet]. Bethesda (MD): National Library of Medicine (US); 2009 Sep-. 4, Metathesaurus - Original Release Format (ORF) [Updated 2021 Aug 20]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK9682/
- Verma, A. (2020, March 18). PyTorch [Tabular] —Multiclass Classification. Retrieved June 22, 2022, from https://towardsdatascience.com/pytorch-tabular-multiclass-classification-9f8211a123ab.
- Wood, T. (2019, May 17). *F-score*. DeepAI. Retrieved July 22, 2022, from https://deepai.org/machine-learning-glossary-and-terms/f-score

Thank You