DATA MINING – PROJECT CHECKPOINT

ORIN BROWN

PROBLEM STATEMENT

- Text classification
  - The goal of this data mining project is to develop a classification model that accurately categorizes new articles into predefined topic categories. Given a vast amount of unlabeled news articles, the objective is to create a reliable system that can automatically assign appropriate categories to incoming articles based on their content.
  - Investigate the prediction stability of text classifiers. Prediction stability refers to the consistency and reliability of the classification results obtained from text classifiers when applied to different samples or variations in the input data. It aims to investigate the robustness of text classifiers and examine how consistent their predictions are across different instances of the same problem.

RELATED WORK

- Traditional models
  - Bag-Of-Words (BOW), N-gram, Term Frequency-Inverse Document Frequency (TF-IDF), word2vec, and Global Vectors for word representation (GloVe).
- Deep learning models
  - Sentiment Analysis (SA), Topic Labeling (TL), News Classification (NC), Question Answering (QA), Dialog Act Classification (DAC), Natural Language Inference (NLI), and Relation Classification (RC).
- Excellent and current overview on text classification work:
  - ACM Transactions on Intelligent Systems and Technology Volume 13Issue 2 Article No.: 31pp 1–41
  - https://doi.org/10.1145/3495162

RELATED WORK

- Term Frequency - Inverse Document Frequency (TF-IDF)
    - Interpreting TF-IDF term weights as making relevance decisions
        - https://dl.acm.org/doi/10.1145/1361684.1361686
    - TF-IDF Keyword Extraction Method Combining Context and Semantic Classification
        - https://dl.acm.org/doi/10.1145/3414274.3414492
    - TF-IDF uncovered: a study of theories and probabilities
        - https://dl.acm.org/doi/10.1145/1390334.1390409
    - Class Specific TF-IDF Boosting for Short-text Classification: Application to Short-texts Generated During Disasters
        - https://dl.acm.org/doi/10.1145/3184558.3191621
    - An Improved TF-IDF algorithm based on word frequency distribution information and category distribution information
        - https://dl.acm.org/doi/abs/10.1145/3232116.3232152

RELATED WORK

- Deep learning text classification
  - Text Classification Based on Graph Convolution Neural Network and Attention Mechanism
    - https://dl.acm.org/doi/abs/10.1145/3573942.3573963
  - Effective Media Traffic Classification Using Deep Learning
    - https://dl.acm.org/doi/abs/10.1145/3314545.3316278
  - Comparing Methods of Text Categorization
    - http://uu.diva-portal.org/smash/get/diva2:1275337/FULLTEXT01.pdf
  - Deep Learning Based Text Classification: A Comprehensive Review
    - https://arxiv.org/pdf/2004.03705.pdf

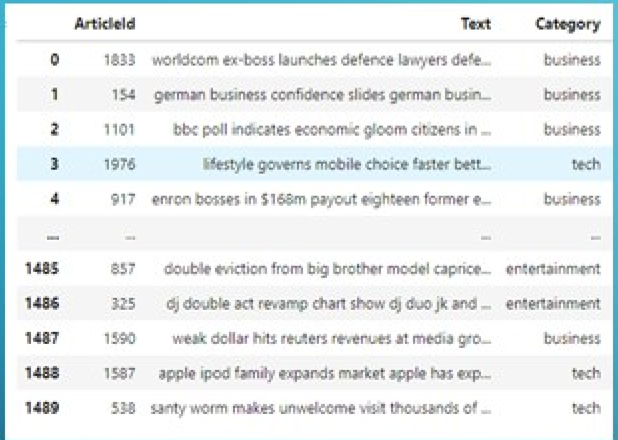
PROPOSED WORK

- Use one traditional model and one deep learning model.
  - As indicated by the previous related work slides, I have decided to use a Term Frequency - Inverse Document Frequency (TF-IDF) model along with as yet to be determined classifiers for the traditional model and a Multilayer Perceptron (MLP) for the deep learning model.
- Find most "diagnostic" words / phrases / word pairs, see if they make sense to a human.
  - TF-IDF should be able to provide this, sorting by word score.
- Find distinguishing patterns for different article types.
  - I'm still looking for what kind of characteristics or metrics I could measure for this purpose. Some ideas are: word length, article length, unique word count.

PROPOSED WORK

- Focus on “problem areas” for text classification models:
  - Stability
    - I’m still investigating methods to make sure predictions are stable to small “perturbations” in the text.
  - Interpretability
    - TF-IDF algorithm will help with interpretability.
    - MLP architecture is inherently difficult / impossible to interpret. Perhaps the best I can do is a high-level and generalized understanding of how the architecture processes the documents, e.g. how the number and type of layers affect the flow of information.

DATA



	ArticleId	Text	Category
0	1833	worldcom ex-boss launches defence lawyers defe...	business
1	154	german business confidence slides german busin...	business
2	1101	bbc poll indicates economic gloom citizens in ...	business
3	1976	lifestyle governs mobile choice faster bett...	tech
4	917	enron bosses in $168m payout eighteen former e...	business
...	...	...	...
1485	857	double eviction from big brother model caprice...	entertainment
1486	325	dj double act revamp chart show dj duo jk and ...	entertainment
1487	1590	weak dollar hits reuters revenues at media gro...	business
1488	1587	apple ipod family expands market apple has exp...	tech
1489	538	santy worm makes unwelcome visit thousands of ...	tech

- Source
  - I’ve identified a website with a large number of different data sets. Many have already been processed and prepared for analysis.
  - https://ana.cachopo.org/datasets-for-single-label-text-categorization
- Warehousing
  - All the data sets are in the range of 10’s of MB, so I will have no problem storing them locally or on github while developing the project.

EVALUATION

- Accuracy
  - The primary metric I will use to judge model performance will be the accuracy of the text classification.
- Prediction stability
  - How the model responds to added text, either significant or random stop words.
- Runtime
  - Speed of training and prediction.
- Interpretability
  - A model will be rated higher if it produces some kind of intermediate output that is interpretable by humans. For instance, a list of the most “diagnostic” words or phrases.

TIMELINE

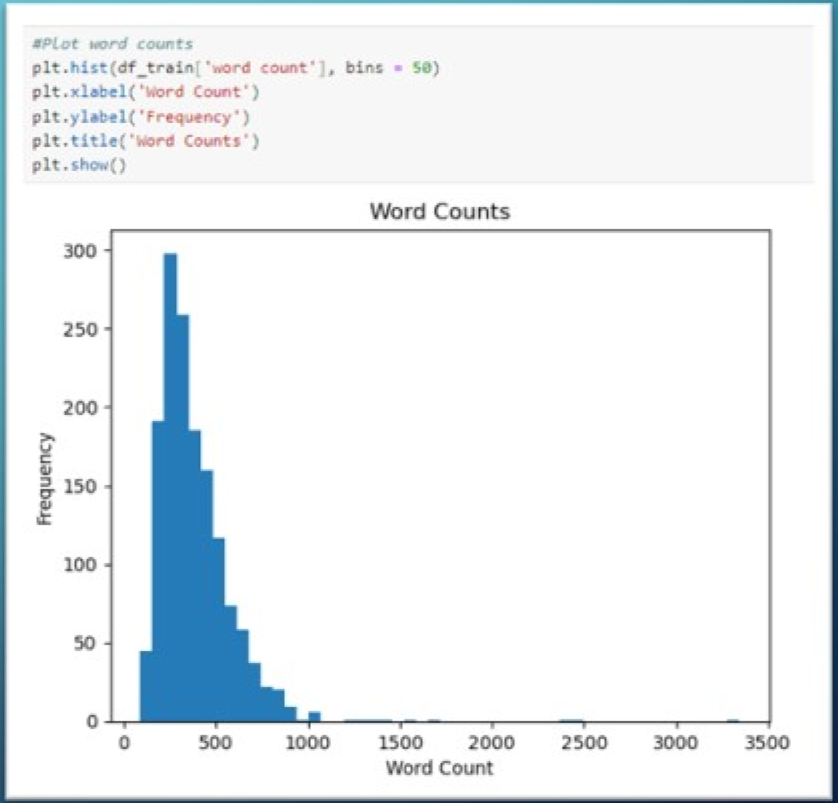
- There are roughly six weeks left in the term at time of writing, so I propose to divide the tasks as follows:
  - Data gathering – 1 week
    - This has proved to be relatively accurate. I feel that I have enough data to train the models on. I may have come back to this step later if that proves not to be the case as I begin modeling.
  - Research on models – 1 week
    - This has run over its original time allotment, which I suppose should not be surprising. There are a seemingly endless number of different model architectures, and more tweaks and permutations on those.

TIMELINE

- EDA and data cleaning – 1 week
  - The data I have found is clean for the most part, and small enough in size that warehousing is not an issue. EDA is ongoing.
- Modeling and model tuning – 2 weeks
  - I anticipate that this will run over its allotted time.
- Analysis and report writing – 1 week
  - I plan to write and do preliminary analysis as I am building the models, so hopefully I can reclaim some time here.
- This is not a hard and fast division of tasks, as I'm sure they will blend into each other, but more of a general prediction as to how I will spend the remaining time.

EDA

- I have begun EDA, first by combining the two datasets, and then by performing preliminary data exploration and cleaning.
- Exploration:
  - Data inspection, preliminary graphs
- Data cleaning:
  - Remove stop words, punctuation, null values.



```
#Plot word counts
plt.hist(df_train['word count'], bins = 50)
plt.xlabel('Word Count')
plt.ylabel('Frequency')
plt.title('Word Counts')
plt.show()
```