**Fake News Detection & Classification:**

Limitations from previous work:

* No explicit demonstration of feature extraction, how they were selected or used as input (quite the case in Springer journals I found)
* Haven’t seen many that applied XLNet or advanced BERT models yet (at least not the ones that were published in journals, one journal that used XLNet that was submitted to ArXiV)

Some important features considered:

* Content and social context-based features
* Semantic, lexical features
* Statement attribute
* Sentiment features
* NER tags as features

My thoughts:

Write a journal by:

Try some more advanced models to try to outperform works that used much-older time series models such as LSTM, GRU.

***Preprocessing:***

**Stemming & Lemmatization:**

Stemming : Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language.

Further reference: <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>

**Journals on Excel file:**

The ones I thought could be good starting point, considering the algorithms I am familiar with, level of journal:

* Fake Detect: A Deep Learning Ensemble Model for Fake News Detection
* Analyzing and distinguishing fake and real news to mitigate the problem of disinformation
* FNDNet – A deep convolutional neural network for fake news detection
* DeepFakE: improving fake news detection using tensor decomposition-based deep neural network

**Points to note:**

Proposed model in “Fake Detect: A Deep Learning Ensemble Model for Fake News Detection” outperformed the one in “An ensemble machine learning approach through effective feature extraction to classify fake news”

**Datasets:**

FNC-1, FakeNewsNet, LIAR

**Prospective Ideas and important points:**

* Spatiotemporal information is something crucial to consider: Spatial (location-based relation between information (CNNs), Temporal (RNNs)
* Applying filters in CNN allows the model to identify spatial patterns (shapes, objects, edges, etc.). Now the filters would be holding the spatial information of the pattern they represent.
* The convolutional calculation takes care of how strongly the spatial relations between the information in the filter correlate/match with those in the input data.
* While CNNs are generally at identifying objects, it has a drawback of being insensitive to potential critically translational relations. Hence, capsule networks were proposed.
* Oversampling and under-sampling can be considered in case there is a big imbalance in dataset in terms of classes

**Useful Reference for text preprocessing:**

* Things to watch out for in texts:
  + nationalities, such as U.S.
  + grades such as K-12

[python - How to count uppercase and lowercase on pandas dataframe - Stack Overflow](https://stackoverflow.com/questions/49230262/how-to-count-uppercase-and-lowercase-on-pandas-dataframe)

[Text Analysis & Feature Engineering with NLP | by Mauro Di Pietro | Towards Data Science](https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d)

Some fundamental, essential terms:

* N-gram

Initialy tried out models:

* SVM + Tf-IDF, SVM + Word2Vec
* CNN + BERT embedding (Jiwi)
  + In NLP, Conv1D is generally used instead of Conv2D
  + In “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection”:
    - Kernel sizes : (2, 3, 4), each 128 filters
    - Dropout of 0.8
    - No L2 penalty applied
    - Batch size: 64

BERT Embeddings:

* Issue: can handle a sequence length of up to 512
* References:
  + [In BERT, what are Token Embeddings, Segment Embeddings and Position Embeddings? – MachineCurve](https://www.machinecurve.com/index.php/question/in-bert-what-are-token-embeddings-segment-embeddings-and-position-embeddings/)

***Attention Mechanism:***

[Attention in Neural Networks. Some variations of attention… | by Mahendran Venkatachalam | Towards Data Science](https://towardsdatascience.com/attention-in-neural-networks-e66920838742)

Self-Attention Mechanism (Transformer: Attention is all you need)

<https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec>

***Potential Base papers:***

1) “An ensemble machine learning approach through effective feature extraction to classify fake news”

* Data: LIAR
* Use of 26 linguistic features and not the text content itself (no use of embedding)
* Model that is comprised of: Decision Tree, Random Forest, and Extra Tree Classifier

2) “A novel self-learning semi-supervised deep learning network to detect fake news on social media\_2021”

* Data: FakeNewsNet
* Uses only text content seemingly
* Model” Bi-LSTM with confidence-function layer

3) “Analyzing and distinguishing fake and real news to mitigate the problem of disinformation”

* Data: FakeNewsNet
* Uses only text content seemingly
* Model: GRU, LSTM, RNN

4) “Fake Detect: A Deep Learning Ensemble Model for Fake News Detection”

* Data: LIAR
* Used 3 experiments, for experiment 1 the authors only used the text content (statement)
* Models: Bi-LSTM, Bi-GRU

5) “Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)”

* Data: FNC-1
* Model for experiment 1: CNN + LSTM with PCA

6) “Fake news detection using deep learning models: A novel approach”

* Data: FakeNewsNet Politifact text data with data collected by authors (30 tweets for each data)
* Models : CNN, LSTM, Bi-LSTM, CNN + LSTM, Bi-LSTM + LSTM, CNN + LSTM with Attention, CNN + Bi-LSTM with Attention, Embedding with Glove

7) “Transfer Learning from Transformers to FNC-1 Task”

* Data: FNC-1 data with ARC
* Models : BERT, XLNet, Roberta each with different vector representation

8) “BerConvoNet\_A\_deeplearning\_framework\_for\_fakenews\_classification”

* Data: Fake News Net
* Models : BERT Embedding with a CNN architecture that used multiple kernels of different sizes to create various feature maps, which when concatenated passes through FC.
* Some base models:
  + Self-Attention layer + ELMO ([A multi-label text classification model based on ELMo and attention (matec-conferences.org)](https://www.matec-conferences.org/articles/matecconf/pdf/2020/05/matecconf_cscns2020_03015.pdf))

Some things about DL to keep in mind:

* Epochs: number of times we are feeding entire training data to the model
* In pytorch training, we start looping with epochs and inside of it we train the model with the data in batches
* Iteration: Total Data size / Batch size It is basically how many times it takes to feed the whole data in batches

***Some important things to consider for proposal model:***

* Our data texts are different in length. Hence, a CNN model that can process text with variable length could be promising
  + Self-attention has been useful for text classification tasks
  + Visualization of explainable Attention outputs (visualize demonstrate which parts of the text made it labeled as “Fake News”)
* Other key words: Hierarchical attention, sequential network, scalar attention, Bidirectional RNNs
* Some other base models we considered: CNN + RNN (LSTM or GRU), RoBERTa + RNN (LSTM or GRU), Bert Embedding + CNN (I made code for this, would need Tatha to check whether it was done properly)
* Could leverage on feature projection (which in previous work gave higher improvements than when applied to RNNs, including LSTM)
* Use of contextual information (sth not a lot of previous researches have considered)
* Perhaps could try RoBERTa (as feature extractor) with FP-Net

Other useful journals to consider for proposal model:

1) “Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier”

* Data: not the ones we are using
* Uses only text content
* Source: [Full article: Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier (tandfonline.com)](https://www.tandfonline.com/doi/full/10.1080/08839514.2019.1661579)

Some of the code issues:

Bert Embedding with LIAR training dataset gave OOM error:

