<https://ieeexplore.ieee.org/document/10065317>

Computation Analysis for Identifying the Protagonist and Antagonist and their Sentiments in Harry Potter Books

* used sentence level VADER (a type of sentiment analysis) to determine the protagonist, antagonist, and neutral character
* They removed numbers since that doesn’t affect sentiment, and also removed stop words, punctuation and spaces before and after sentences
* Made everything lowercase
* They preprocessed the data using tokenization and grouped related words using lemmatization
* Character considerations:
  + the different names Voldemord has (He Who Must Not Be Named, the Dark Lord, You Know Who)
  + Only considered first names for the weasleys since can’t distinguish who is being talked about otherwise
  + Only considered first names for professors like Snape or Dumbledore cuz again hard to know who is being talked about

<https://ieeexplore.ieee.org/document/10216289>

The Importance of Context for Sentiment Analysis in Dialogues

* Used 2 datasets, one from a telecom company and one from twitter
* Used binary classification for sentiment (negative and non-negative)
* Considered seven levels of context: the three utterance-levels for the customer (3), the three utterance-levels for both speakers (3), and the current utterance for the service (1)
* Their results:
  + When does the inclusion of context improve the model’s performance compared to excluding previous utterances? For any of the datasets, the BERT-based classifier (BERTimbau), pre-trained using differents types of text in Portuguese, performs better or equally when considering the context) Furthermore, in the TeleComSA dataset, the RoBERTa classifier, pre-trained in multilanguage tweets, achieved the highest performance when changing from a no-context level to a two-sentence context level, improving its score by 29 percentage points. If the dataset used contains more variability, our results suggest that a Random Forest model can also benefit from the use of more contextual information.
  + Overall, when considering context, it is likely that the classification task will improve if we provide the models with more information

<https://ieeexplore.ieee.org/document/10335115>

Implementation of Burmese Language News Classification System by Using SVM and LSTM Machine Learning Algorithm

* Preprocessing, features extraction, SVM and LSTM train models, then classifier model predicts news classification
* Preprocessing
  + Data cleaning: removing noise such as having too many typos or misspellings, having many numbers and punctuations, having many emojis, usernames and links, containing lots of contractions and hyphens, repetition of letters.
  + Tokenization
  + Removing stop words
* Feature extraction: using TFIDF
* Model training:
  + SVM
  + LSTM
* LSTM slightly more accurate, but in real time testing both are slightly off since some of their categories are kinda related

Carl’s random reading list:

[Study of Dependency on number of LSTM units for Character based Text Generation models | IEEE Conference Publication | IEEE Xplore (utoronto.ca)](https://ieeexplore-ieee-org.myaccess.library.utoronto.ca/document/9132839)

Study of Dependency on number of LSTM units for Character based Text Generation models

* Increasing the number of LSTM cells generally improves prediction but above a certain threshold results are poorer.
* Experiment generates C programming language instead of English language to have less vocabulary
* LSTM handles vanishing/exploding gradients better than plain RNN cells
* Experiment has 2 models, one with 2 LSTM layers that are fed the same input, one LSTM layer inputs into the other and then into a Dense layer for output and another similar model with an add Convolutional layer in front to do feature extraction
* Datasets include C programs without comments to avoid problems from using English language, and another with funcation and variable names removed and tokenized
* Experiment also varies the number of LSTM units
* Increasing the number of LSTM cells allows the model to perform better until a certain point where overfitting occurs and there are semantic relationships between characters. Decreasing the word count for the dictionary of words in the data reduces the problem of overfitting and allows an average number of LSTM cells to perform better.

<https://www-engineeringvillage-com.myaccess.library.utoronto.ca/app/doc/?docid=cpx_186d2fd818a3db1457dM671310178165150>

Efficient Method for Personality Prediction using Hybrid Method of Convolutional Neural Network and LSTM

* “While CNN is good at extracting features that are independent of time, LSTM is better at capturing long-term dependencies. combination of features for personality prediction, the LSTM-CNN model is superior to the individual models.”
* Lots of psychology behind the idea of a person’s personality
* Social networking has increased as a means of communication
* Combining the large amount of textual data created everyday and the increase in computing power available models are able to predict human behaviour very well
* CNN is not suited for time series information (like text that has high contextual dependence) and RNN can have disappearing gradients, aims to use CNN and LSTM to out-perform other models
* Preprocessing included convolution filter, mairesse baseline feature-set, only using non-neutral sentences, and vectorizing words using word2vec
* Treated social networks like a graph with relationships between users
* 2 LSTM layers (temporal features) -> 2 convolutional layers (spatial features) with max-pooling layer in between -> batch normalizaiton -> global average pooling -> output dense layer with softmax
* The CNN-LSTM hybrid model performed better than the CNN and LSTM models on their own

<https://www-engineeringvillage-com.myaccess.library.utoronto.ca/app/doc/?docid=cpx_447d29eb17ed4ed5a28M6f161017816328>

Predicting personality traits with semantic structures and LSTM-based neural networks

* Law enforcement, institutions, human resources, advertising and preliminary stages of psychology can all use personality traits as a parameter to make decisions about a person
* Internet grants access to a large amount of information that can be searched for repeatedly
* Social media content may or may not be consistent with a person’s personality, more akin to a persona, a consideration to consider
* Preprocessing tasks in lit review:
  + A. Deleting punctuation marks
  + B. Deleting numbers
  + C. Transforming text to lowercase
  + D. Clearing stop words
  + E. Deleting excess spaces
  + F. Root reduction (stemming, lemmatization)
  + G. Word count
  + H. LIWC features
  + I. Content categorization
  + J. Extracting information from content
  + K. Feature extraction from image
  + L. Face detection from picture
  + M. Filtering non-text objects
  + N. Feature extraction from text
  + O. Feature selection
  + P. Conversion of different languages
  + Q. Deleting special characters
  + R. Word spelling correction
  + S. Vectorizing
  + T. Converting non-text objects to meaningful text
  + U. PoS tagging (Part-of-speech tagging)
  + V. Tokenization
  + W. Word type inference (verb, object, etc.)
  + X. N-grams
  + Y. Converting emoji and emoticon to meaningful text
  + Z. Positive-negative tagging
* The analysis depends on the structure of the social media platform itself as that affects the dataset used. There is also a concern of the diversity in these datasets.
* Authors focused on twitter data
* Tweets containing a chosen 13 words with both positive and negative meaning were collected over the spam on 14 minutes, users from this pool of tweets were selected if they shared tweets publicly and also wrote English content and had a total number of tweets in a certain range. This process yielded over 11 million tweets from over 5000 users.
* Tweets were input into the IBM Personality Insight Service
* Number of tweets from a user and word count of a tweet were taken into account
* Vectorization of text based on frequency, or using FastText, CBOW, Skip=Gram
* After testing, swish activation func, adam optimizer, MSE loss, RMSE evaluation were used.
* Bi-LSTM, using LSTM in forward and backward directions
* PAN-2015 English dataset used as benchmark dataset
* After preprocessing they find that Bi-LSTM gave a better result than LSTM
* They found that obtaining all of a user’s tweet information is important to the success rate.
* They’re own optimized pre processing and vectorization process performed better than FastText
* Keep generalization through the preprocessing steps

<https://www-engineeringvillage-com.myaccess.library.utoronto.ca/app/doc/?docid=cpx_6c5c94d9187d8dd43a4M68ba10178165143>

A Plausible RNN-LSTM based Profession Recommendation System by Predicting Human Personality Types on Social Media Forums

* Using linguistic characteristics from social media platforms as data
  + There is a lot of data in many forms from posts, articles, tweets, etc.
* Used MBTI personality model as a way to classify personalities
* RNN compared to KNN, Logistic regression, Naive Bayes, SVM
* Natural language procession, social science, e-commerce, and personal counselling are interested in this and have practical applications for this task
* This experiment’s goal was to identify a twitter users personality based on their social media activity and recommend a suitable profession for the user
* Using MBTI personality dataset from kaggle for training and testing and data from social media for testing
* Preprocessing of data includes removing separators, urls punctuation, numbers, changing all text to lower case, Lemmatization (collecting various forms of a words to be analyzed as a single item), and vectorizing the text
* K-fold cross-validation used for train-test split
  + Change the size of a input/target data and which chunk of data is the target
  + Change value of k and check performance of model
* LSTM to avoid vanishing gradients
* Sigmoid activation, tanh loss, adam optimizer, nomalized data, softmax activation
* Classification is broken in 4 binary tasks akin to the 4 letters in an MBTI personality type
* 97.81% accuracy on MBTI test data, RNN-Prs did better than other learning algorithms

Joaquin list:

Sentiment analysis using product review data

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2>

* Looked at Sentiment analysis for Amazon reviews
* Used a Part-of-speech (POS) tagger to tokenize the words within each sentance to their role in the sentence. This was helpful since nouns and pronouns usually don’t indicate sentiment
* Specifically identified negative phrases formed by a positive adjective/adverb combined with a negative prefix (i.e. not, never etc).
* Calculated a sentiment score value for each word and phrase token. This was linked to the number of stars in the review.
* They turn the data/tokens into a feature vector by turning it into a binary string, and then parsing it through the built in python has function
* Put its actual model information in the methods section in which it just used a python package. Didn’t use Neural Networks and got like 85% accuracy. Since the paper is from 2015 it makes sense they wouldn’t have used Neural Nets
* Compared the different python packages

Thoughts:

* Converting words to tokens to feature vectors seemed like the most important part of the paper as they went into detail into that, rather than the actual models being used.
* Makes me curious as to whether understanding morel linguistics would be helpful in NLP
* Paper sights a lot of other papers for its math and method

VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

<http://eegilbert.org/papers/icwsm14.vader.hutto.pdf>

* Its named Vader because the lexicon its beating is called Luke lmao
* Sorts tweets into negative, neutral, and positve classes
* VADER Is open source
* Luke was previously really good at analysing text and was used by psycholiogist and other researchers in predicting social media sentiments such as depression
* Paper outlines a lot of differnet lexicons from the past that it will later compare Vader to
* One set of models is just binary decision. One set does intensity of good or bad sentiment, and one set that looks at the word in context rather than just on its own
* The Vader method/model requires no training data. They mention machine learning models in the paper but they kinda frown upon them because of the black box nature
* “Training data” is a comibnation of the regular words from the previous lexicons, and some emoticons, which are all labeled through an Amazon service for remote human labour, on a scale of -4 to 4
* The five rules applied no matter which MI model they were using
* The last sentence in the paper seems to push for more humanity and less MI which is ironic considering the name of their model

Convolutional, Long Short-term Memory, fully connected Deep Neural Networks

<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43455.pdf>

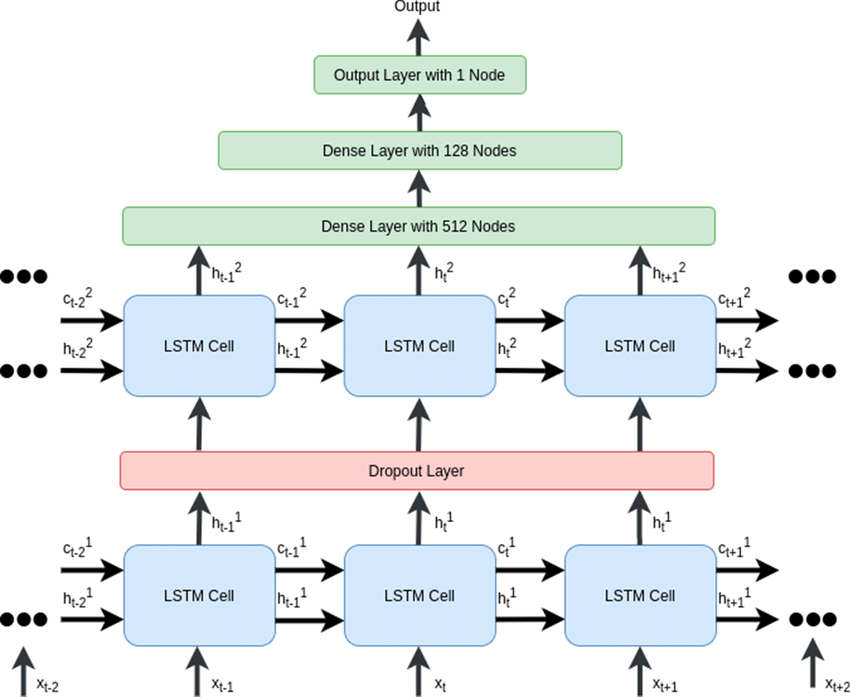
* Combining LSTM, CNNs, and DNNs
* Source [4] goes over LSTMs. (R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to Construct Deep Recurrent Neural Networks,” in Proc. ICLR, 2014.)
* “CNN layers, which reduce variance in frequency of the input, before passing this to LSTM layers to model the sequence temporally.”
* CLDNN has a 5% improvement over LSTM in voice search problems
* Section 2 talks about the CLDNN Architecutre
* 2 cnn layers, 2 lstm layers, linked with a linear layer to reduce dimensionality. Then output is passed into DNN layers for the final output
* Trained from noisy sound audio.
* Tested each model separately for baselines, before combining them
* Section 4 shows the comparisons between the different architecture combinations
* At the end the Word Error Rate of the CLDNN was the best

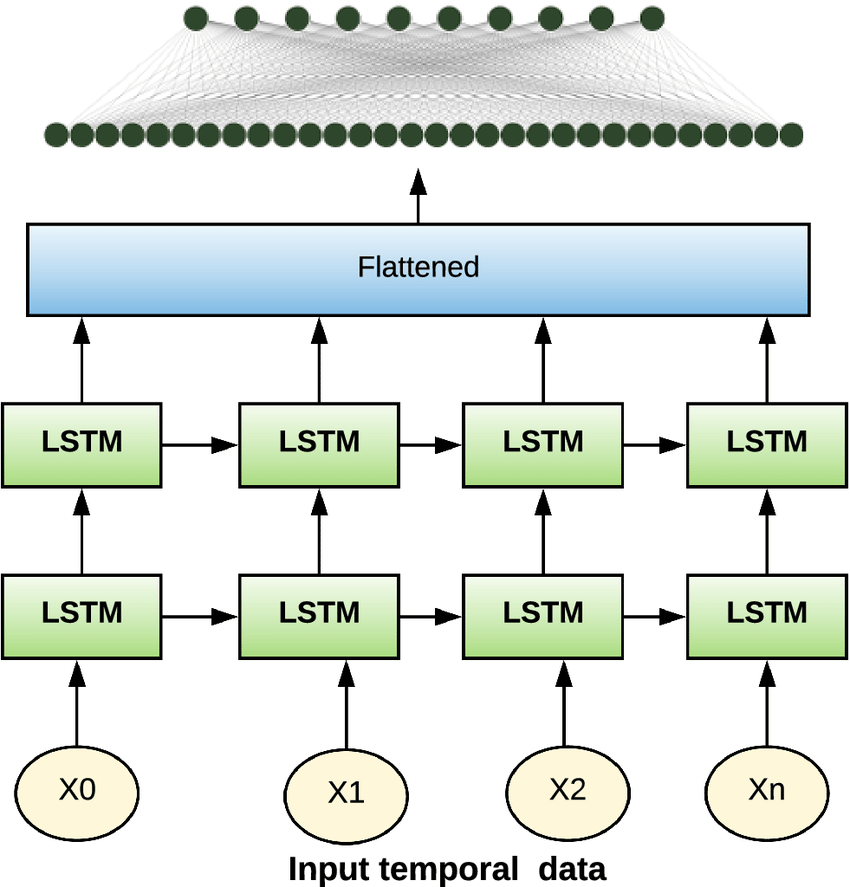
Meeting notes:

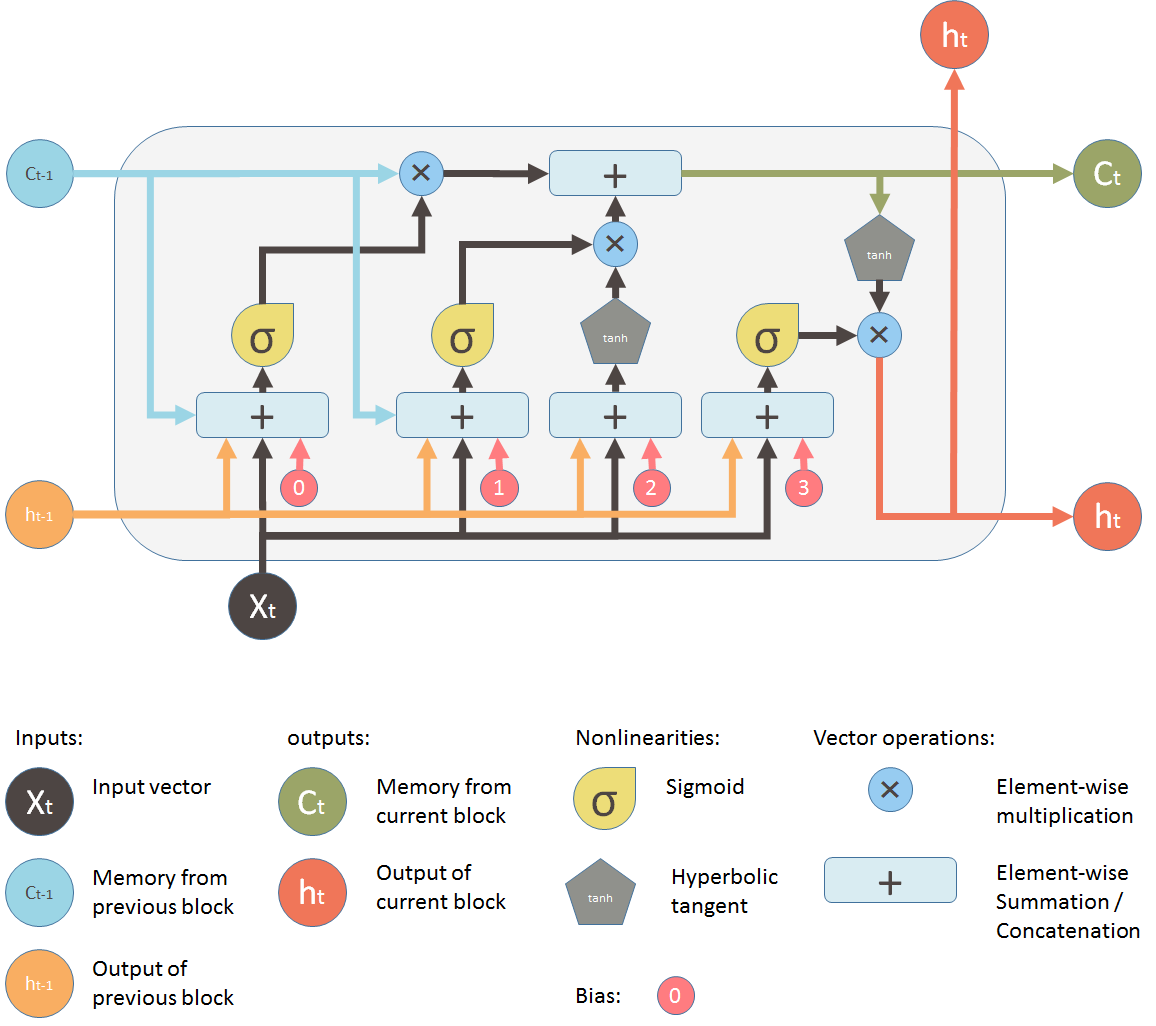
* Already a paper using Harry Potter data
  + Preprocessing can be similar
* Context paper
  + More context is better
* LSTM Layers paper
  + Model based
* CNN LSTM model
* Bi-LSTM
* MBTI Model
  + Four binary decisions
  + Can also compare pairs of houses
* 2 more articles
* 1 more article each
* Keep Keywords

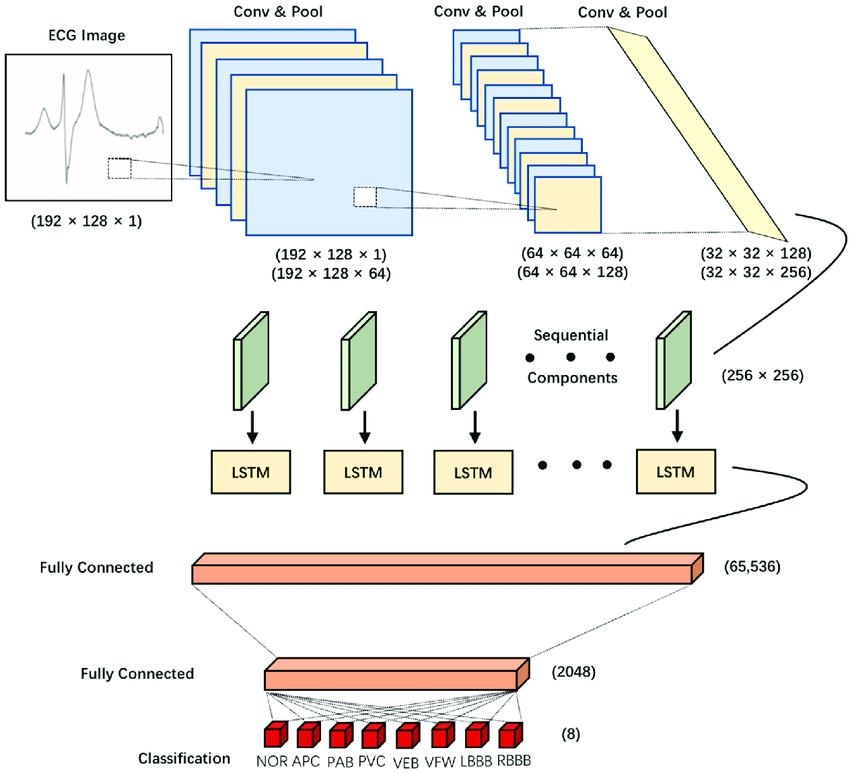
Pick a model:

* CNN-LSTM?
* Stacked LSTM?
* Diagram -> Carl









Sections:

* Introduction -> Sandhi
* Abstract -> Carl
* Problem Statement and Model -> Joaquin
* Describe Data Set -> Joaquin

Together:

* Bibliography
* Lit Review