Predicting Audience Reaction to a Political Speech: Applying a Compound Architecture to Efficiently Process Context

Samuel Lurye EE '20
slurye@stanford.edu

Motivation

Speeches are often intended to provoke an **emotion or action** in their audience, so a predictive analysis ahead of delivery can be invaluable to successfully driving people towards the intended goal. Audience reaction is the **simplest indication of more complex internal feelings**. The purpose of this project is to create a model to predict a sentence-level audience reaction to a written speech to **provide a heuristic for the effectiveness of the speech**.

Data

- The data consisted 3, 618 political speeches, from 197 different speakers totaling 7, 901, 893 words in total.
- There are a total of 14 unique tags throughout the speeches, creating a tag density of 0.0084. [Figure 1]
- Tags were grouped into four categories to simplify classification process. Distribution can be seen in [Figure 2]

SINGLE TAGS	
{APPLAUSE}	46310
{LAUGHTER}	14055
{AUDIENCE}	1803
{BOOING}	756
{SPONTANEOUS-DEMONSTRATION}	313
{CHEERS}	234
{SUSTAINED APPLAUSE}	97
{STANDING-OVATION}	51
MULTIPLE TAGS	
{LAUGHTER ; APPLAUSE}	1579
{CHEERS ; APPLAUSE}	837
OTHERS	47
SPECIAL TAGS	
{AUDIENCE-MEMBER}	999
{COMMENT}	787
{OTHER-SPEAK}	404
GROUPED TAGS	
POSITIVE-FOCUS TAGS	49275
IRONICAL TAGS	15660
NEGATIVE-FOCUS TAGS	1147

Figure 1- List of Tag Frequencies Across all Speeches

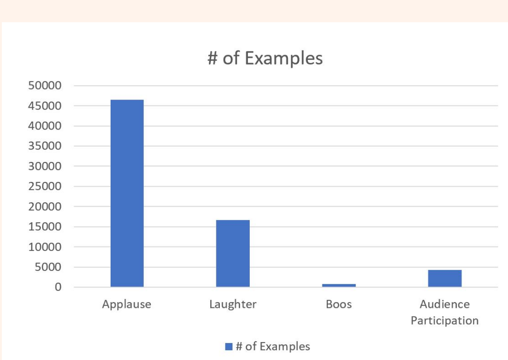

Figure 2- Distribution of Grouped Tags

Model

LSTM-CNN

Sub-section [1] is a standard LSTM-CNN pairing common in many sentence classification tasks. Each word in a target sentence into an **embedding using Word2Vec** and feed through a bi-directional LSTM to **capture the long-term dependencies** in the sentence structure. The forward and backward hidden states for each cell are concatenated and **passed as inputs to five independent CNN** layers, each with a different kernel size (varying from 2 to 6). The CNN independent layers with max-pooling are designed to **extract features from every part of the sentence and catch different sized interdependencies.**

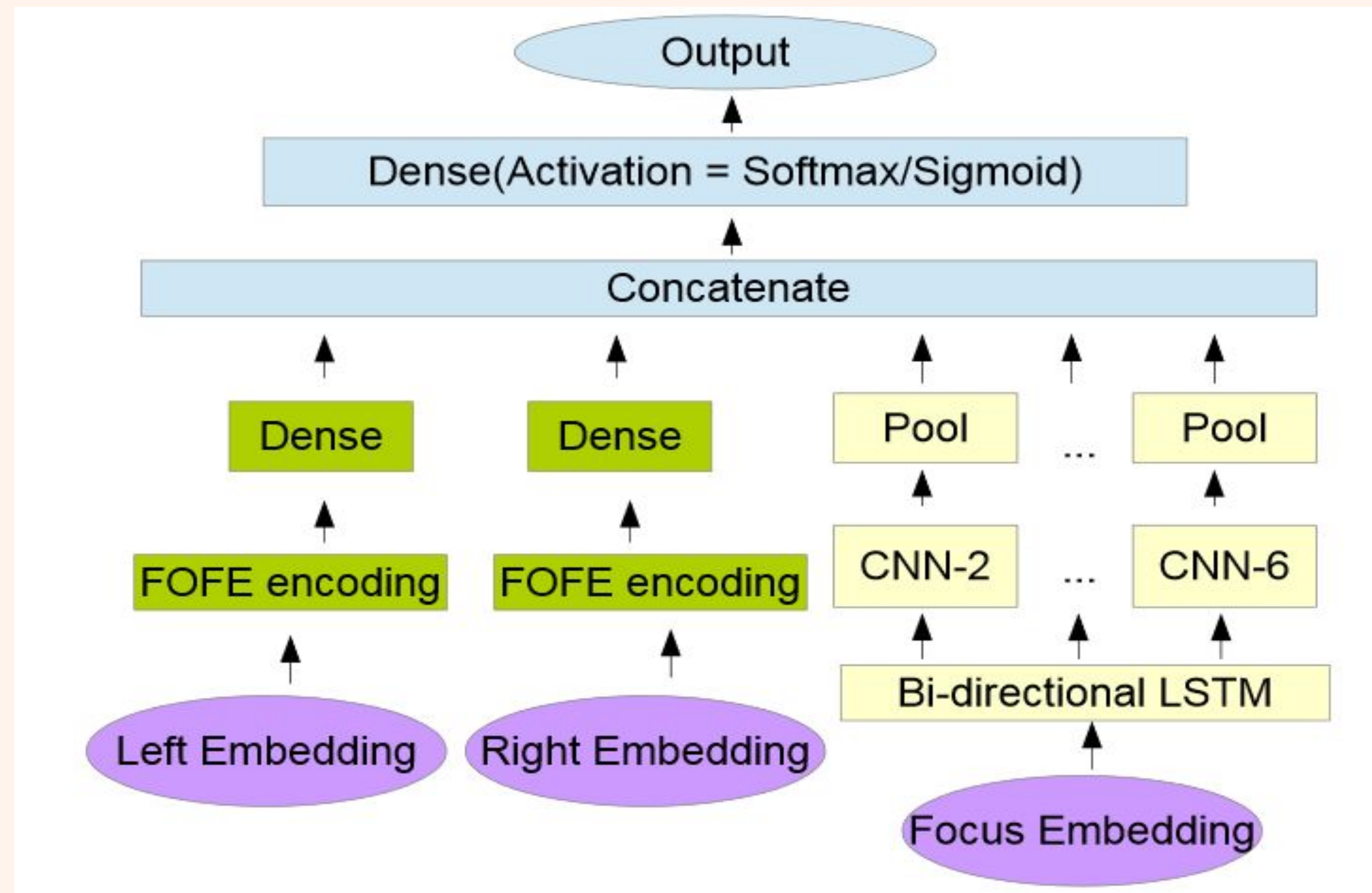

Figure 3 - Block diagram showing model, inputs, outputs, and cost

FOFE Encoding

The embedding z for a sentence $(x_1, x_2, \dots x_U)$ is initialized to $z_1 = x_1$,then calculated recursively for $u \in 2 \dots U$ $z_U = \alpha * z_{u-1} + x_u$ parameter $\boldsymbol{\alpha}$ is the forgetting factor. This puts heavy bias on sentences more local to the target sentence while keeping the importance of all words within the sentence the same.

Results

Hyperparameter Decisions

Hyperparameter	Choice	Tested
LSTM Hidden Size	100	50-300
CNN Stride	2	1-5
Learning Rate	0.001	0.001 - 0.01
Dropout Rate	0.3	0 - 1
CNN Pooling	Max	Max-Mean
CNN Output Channels	10	5,10,20,50

Table 1- Selection of hyperparameters in the model

Context vs. Accuracy

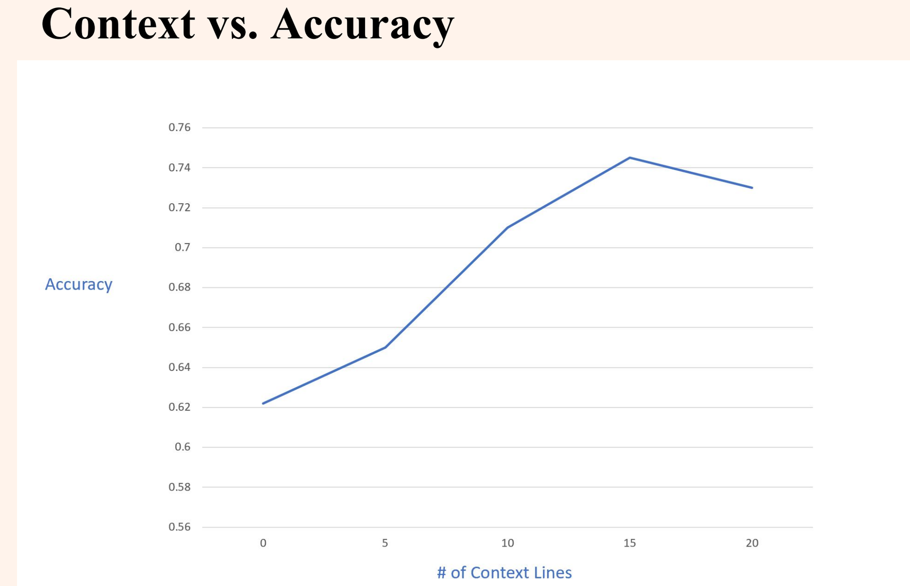

Figure 4- Test accuracy is plotted vs. # of lines of context used

Model Performance

Model	Accuracy	Precision
CNN (baseline)	0.604	0.62
LSTM-CNN	0.622	0.701
C-LSTM-CNN (best)	0.745	0.76

Table 2- Performance of models

The LSTM-CNN model was a minor improvement on the baseline model, but did not show the significant gap that the architecture improvement would suggest. We ran four different version of the C-LSTM-CNN model, varying the total leading and lagging context used. The values in the table represent the optimal context of our experiment, 15 lines.

Discussion

- **Fewer hidden units** in the LSTM cells result in **comparable** performance at significantly **shorter** training periods.
- There appeared to be a **saturation threshold** above which adding additional context only diluted the predictive value of the FOFE encoding
- Dropout had **no meaningful effect** on architecture because of its compound nature.
- Multiple fully-connected layers in between LSTM cells improved performance.
- Our model was able to **outperform** the naive **CNN** and **LSTM-CNN** models.

Future Work

- Gather **significantly more data**. The amount of data points we had for individual classes is small for modern deep learning algorithms.
- Update the encoding algorithm. FOFE encoding appears to lose significance when context becomes too big.
- Develop test sequences of significant length (10 - 20s) and test extended model performance over them.
- Train with **lower learning rate** and higher hidden layer size on more powerful computers.

References

1. Guerini, M., Giampiccolo, D., Moretti, G., Sprugnoli, R., Strapparava, C. (2013). The New Release of CORPS: A Corpus of Political Speeches Annotated with Audience Reactions. Shaping Minds and Social Action,86-98.
2. Sainath, Tara et al. 2015. Convolutional, long short-term memory, fully connected deep neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing.
3. Song, Xingyi et al. "A Deep Neural Network Sentence Level Classification Method with Context Information" ACM, 2018.