Emo-Context

IRE Major Project: Team 18

Problem Statement

Given a textual user utterance along with 2 turns of context in a conversation, classify the emotion of user utterance as happy, sad, angry or others

Initial Brainstorming

Reading through previous works, such as MSR India's paper, *A Sentiment and Semantics Based Approach for Emotion Detection in Textual Conversations*, led us to believe that deep neural networks are more suitable to solving such a problem than conventional machine learning classifiers like SVMs. This is due to the highly complex nature of the problem and the data itself.

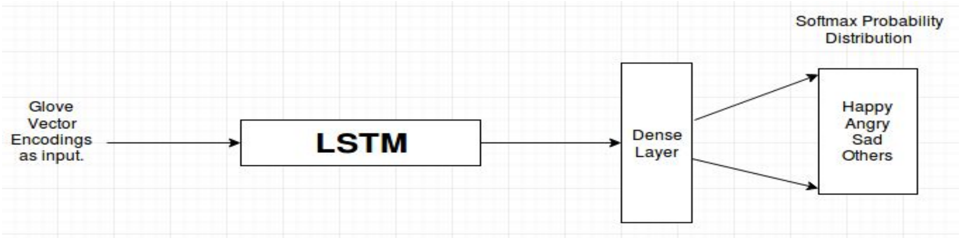
Thus, we decided to apply our efforts into deep learning models.

Approach

- Representation of the text in a format suitable for application of statistical methods
- Using feature engineering to emphasize unique attributes of the textual units that may be of consequence
- Experimentation with varying values for hyperparameters, and various architectures; including models with CNNs and RNNs

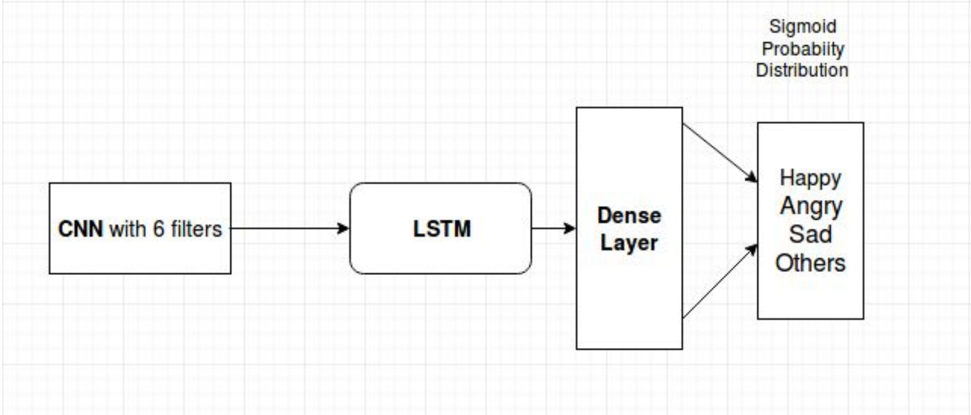
Model 1 (F1 = 56.73%)

1. A simple LSTM followed by a dense layer, which utilizes a sigmoid activation function to give class-wise probabilities
2. 100 dimensional GloVe embeddings
3. LSTMs help to retain some contextual information
4. We treat this model as our baseline



Model 2 (F1 = 55.xx%)

1. LSTMs, due to their recurrent behavior take time to train; CNNs are faster in comparison
2. CNN with 6 filters, a max-pool, and then an LSTM followed by a dense layer and finally a four class sigmoid activation layer
3. The scores actually get worse; some possible reasons - the CNN mangles the conversation and actually ends up losing information, and context is lost in max-pooling



Feature Engineering

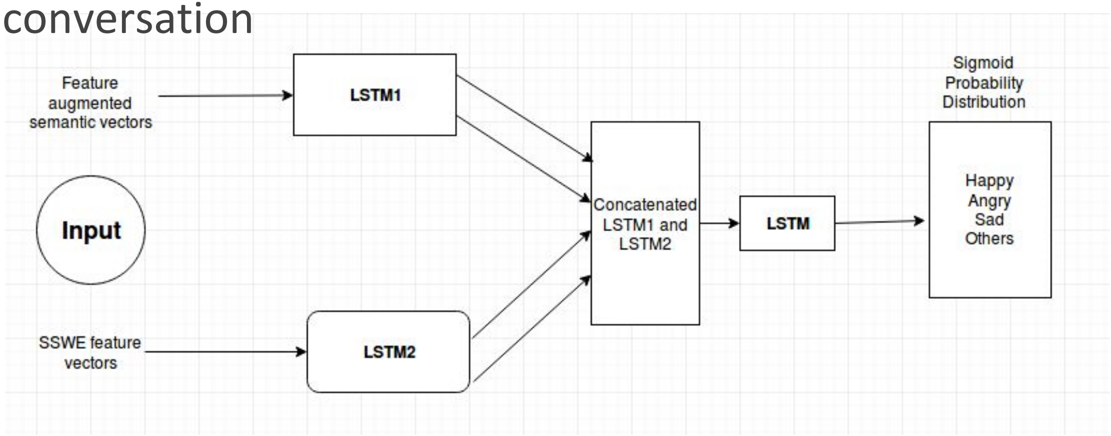
1. Feature engineering to produce the following features per word
    a. Is the word a *hedge*?
    b. Is the word a *factive verb*?
    c. Is the word an *assertive verb*?
    d. Is the word an *implicative verb*?
    e. Is the word a *report verb*?
    f. Is the word an *entailment causing verb*?
    g. Is the word a *subjective verb*?
    h. If yes, is it *weakly subjective* or *strongly subjective*?
    i. Is the *polarity* of the word positive or negative?
2. We append these features to the regular GloVe embeddings to produce what we call *feature augmented semantic word embeddings (FSWE)*

Model 3 (F1 = 59.42%)

1. Replacing GloVe with our own FSWE in model 1 results in greater score by a margin of ~3%
2. Further changing the GloVe embeddings from 100 to 300 dimension raises the score by roughly 1%
3. At this point it is clear that the features have an overall positive effect, and hence are used in every model here on

Model 4 (F1 = 66.09%)

1. **Best performance yet**
2. *Sentiment specific word vectors (SSWE)* feature vectors hold both sentiment and semantic information
3. LSTMs 1 and 2 represent the conversation
4. The final LSTM draws correlations from both, the semantic and sentiment spaces
5. Dense layer extrapolates vectors into probability distribution



Conclusion

Performance seems to be correlated to the following -

- the amount of importance given to the context
- semi-supervization via external feature generation
- hyperparameter adjustments

At the moment we have a number of other ideas that seem worthy of trial; future approaches will also involve more complex architectures including hierarchical models, and manipulation of training data.

Thank You!

Links

- GitHub
- Project Webpage
- Video Demo

References

- *A Sentiment and Semantic Based Approach for Emotion Detection in Textual Conversation, Gupta et al.*
- *Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification*
- *Emotion Detection from Text*