Subvocal Representations of Natural Language for Text Classification

Jason Stock
stock@colostate.edu

Dept. of Computer Science, Colorado State University

Abstract

Accurate classification of text is of paramount importance in natural language processing– enabling automated analysis and insights to the data. Many existing works rely on numerical representations of text to decode written language; however, this does not truly reflect the human reading process. As such, this work introduces a method to computationally emulate subvocalization with text and synthesized audio components. Specifically, we combine the high-level non-linear features of these components with a neural network for text classification. The proposed network gets 64.48% accuracy on a sentiment analysis task, and experimental results are shown to outperform a text-based LSTM, text-based CNN, and audio only approaches. While this work focuses on text classification, our method can be extended to other problem domains where raw text is in a readable format that can be subvocalized.

1 Introduction

The phenomena of inner speech, also known as internal dialog or subvocalization, commonly occurs during silent reading where individuals silently speak or hear the text that is read. Experimental insights drawn by psychologists and neuroscientists suggest that this process can reduce cognitive load and yield greater comprehension and memory of the text [1, 2]. As it relates to spoken words, speech can be characterized with contextual information, *e.g.* rising intonation in a question or change in pitch to emphasize a word, that can affect intelligibility. However, many natural language processing tasks in machine learning only consider the written component to derive meaning or associations. In this work, we seek to join the textual features and generated audio sequences of text-only data to understand how to better model the process of inner speech.

As it relates to similar work [3–6], we may expect the combined model (text and audio components) to perform best, however; these works use ground truth transcribed audio, whereas this work relies on the quality of synthesized audio and the challenges that come with, *e.g.* encoding emotion and pronouncing complex words, which further propagate to our results. To measure the efficacy of our approach, we also compare multiple models with and without these audio components.

The multimodal neural networks proposed within relates to work from Gu *et al.* [3, 4] in that we use a text and audio branch with raw text and spectrogram-based audio representations. However, we explore pre-trained word embeddings and different architecture features to process the data. Han *et al.* uses multi-headed attention with recorded audio, video, and text for sentiment analysis and observe the supplementary features to be useful. Our method uses a simplified text-based neural networks but could be adapted for transformers and similar attention-based models. In all, the primary differentiator with existing work is that we synthesize audio from text-only data.

2 Dataset Details

We evaluate results of our proposed approach using the TweetEval dataset, which consists of several heterogeneous classification tasks of Twitter. Of the seven available benchmarks, we consider emotion recognition [7] and sentiment analysis [8] for a more comprehensive analysis. Both are of the same structure with short text snippets of 250 characters or less with their corresponding labels. The content of these two datasets vary and directly influence the overall sample size. As such, the emotion data has 5052 (3257/374/1421) samples and the sentiment data totals 59 899 (45 615/2000/12 284) samples.

Tweets were collected between 2017 and 2018 and annotated using CrowdFlower. The class distribution are not equal but are partitioned with stratified sampling. Each sentiment dataset contains roughly 18.9%, 45.9%, and 35.1% for positive, neutral, and negative classes, respectively. Similarly, the emotion datasets are partitioned into 41.9%, 23.0%, 8.8%, and 26.3% per class groups for anger, joy, optimism, and sadness, respectively. An example tweet is as follows with mentions and hashtags present:

> *"Once you've accepted your flaws, no one can use them against you - @user #quote #mentalhealth #psychology #depression #anxiety"*
>
> — TweetEval Emotion, *Optimism*

Supplementary to the raw text are generated audio samples as mel-spectrograms from the FastPitch model [9]. This text-to-speech model was trained on 24 hours of single-speaker audio samples from the LJSpeech 1.1 dataset [10]. Without any additional training, we use the optimized transformer-based network to predict pitch contours with the minimally processed tweets as input. We only change the pacing parameters from 1.0 to 1.2 and keep all other arguments as default for inference. Outputs are mel-scale spectrograms (Figure 1) that represent the frequency content of the text over time. Simply, each text sample has a corresponding "image" of the decomposed audio with constant height and variable time duration.

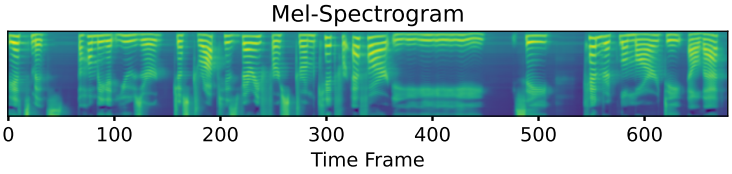

Figure 1: Mel-spectrogram of a positive TweetEval sentiment sample representing the text *"Sun is shining... Bob Marley on the radio... hiya Monday!"*

3 Proposed Architecture

Modeling subvocalization is done with a multi-branch neural network that combines deep embeddings of textual and audio components. Specifically, the network allows for the multiple inputs to be processed individually and then subsequently concatenated by their hidden representations. The joint features follow one fully-connected layer with 10 non-linear units to the output which we use to predict the corresponding class. Details of each branch are outlined in Section 3.1 and Section 3.2.

Weights in the model are randomly initialized and updated via backpropagation using Adaptive moment estimation with $\eta = 0.001$ to optimize the cross-entropy loss between target and predicted classes. Both branches are trained simultaneously with the same loss using a batch-size of 32 maximum samples over 15 epochs. Lastly, we default to using the hyperbolic tangent as the non-linear activation as it performs best in our hyperparameter search.

3.1 Text Branch

The minimally processed text are tokenized on white space with capitalization removed. Experimental tests show training a custom embedding results in lower performance metrics, so we use the Global Vectors for Word Representation (GloVe) embeddings [11] pre-trained on a Twitter dataset of 2B tweets. The 50 dimensional embedding vector has a vocab size of 1.2M and every word in our data is mapped to one of these indices (unknown words are mapped to `<unk>`). As such, the input to the embedding layer (with frozen GloVe weights) is a list of word indices with variable length.

To ensure the model has consistent length input samples, we pad each sentence in a mini-batch with the index of the GloVe `<pad>` token in accordance with the longest sentence in that batch. Thereafter, the sampled word embeddings are input to either a Long Short-Term Memory (LSTM) layer or series of convolutional layers. Having the ability to choose one over the other is to allow exploration on the effectiveness of each layer type for our task.

The LSTM layer uses the conventional gated connections and series of operations. The input size is that of the embedding with a hidden size of 64 in only one layer. Additionally, both the hidden and cell states follow Xavier normal initialization at the start of every mini-batch for training and inference. The major change to the input of this layer is how the embedding vectors are represented. Specifically, we pack the padded sequence to recover the actual samples without the pad embedding which further optimizes the computations. Only the last hidden state from the LSTM propagates forward through the subsequent layers.

If the convolutional layers are specified instead of the LSTM, then we learn weights for filters that convolve over the embeddings. The height of these filters are a specified window size with a width the size of the embedding vectors. After each convolution is a non-linear activation and a maxpooling layer that reduces the sequence length by two, thus keeping the size of the embeddings constant. We repeat this convolving and pooling process four times using eight filters per layer with window sizes of 1, 3, 5, and 5 from first to last. Finally, we take the mean over timesteps to reduce the dimensionality to a constant value that is compatible with fully-connected layers.

3.2 Audio Branch

Similar to the text branch, we have to deal with variable length mel-spectrograms when processing the frequency spectrum of the audio signals. Therefore, we pad each mini-batch sample over time with zeros for a specified length of the longest training sample. These single channel images are normalized between $[-1,\ 1]$ (prior to padding) using the minimum and maximum values in the training data and fed as input to layers of 2D convolutions and maxpooling. The parameters of these layers are common with many computer vision problems and are composed of four convolutional layers with 8, 16, 32, and 64 filters all with 3×3 filter sizes. The resulting activation filter maps are flattened to 1 dimensional vectors. Following are two fully-connected layers with 512 and 64 non-linear units.

4 Experimental Results

We train four separate networks, namely text-lstm, text-conv, audio, and combined, that are evaluated on both test datasets. All models are trained using the same hyperparameters and weight initialization but differ in their architectural structure. Training and inference runs on a single NVIDA RTX 3090 with 24 GB to accelerate computation. These models are relatively quick to train with one epoch of the emotion dataset passing in 0.205 s and 1.660 s for the text-lstm and combined network, respectively. The text-conv model is similar in time to the text-lstm and the audio model is the difference between the combined and text-lstm.

Regardless of the data, these models are prone to overfit as observed by the error curves of training and validation loss (not shown within). Reducing the number of trainable parameters in the text-branch, whether that be in the fully-connected layers or convolutional filters, is the most effective method to reduce overfitting. Experiments with regularization, namely batch normalization and dropout, are effective to generalizing but yield a performance decrease with our setup. Using the previously discussed networks, we find the text branch to be significantly more stable with a smooth approach on a local minimum, whereas the networks with mel-spectrograms are difficult to train. In fact, having too few parameters cause the audio model to make predictions on only one class. Thus, we find a trade-off on the number of parameters that maximize generalization.

Table 1: Total accuracy and macro-averaged F1 score over all classes on the test set of the emotion and sentiment datasets. Results are compared with the four different model types.

Model	Emotion		Sentiment	
	Acc. (%)	F1	Acc. (%)	F1
text-lstm	**64.25**	**0.581**	62.84	0.620
text-cnn	55.80	0.507	48.33	0.217
audio	32.44	0.253	48.24	0.210
combined	63.27	0.548	**64.48**	**0.631**

In comparison to results from Barbieri *et al.* [12], we find our results to be comparable to the bi-directional LSTM model with improvements to sentiment analysis, but fall short of state of the art methods. Table 1 shows the overall accuracy and macro-averaged F1 score over all classes for each network and values in bold represent the top performers. The text-lstm model achieves 64.25% and 0.581 F1 score on the emotion test set with nearly equal precision and recall on all classes but "optimisim". The model incorrectly classifies this class more frequently as "anger" than it does the actual target. We speculate this observation to be closely related to the class imbalance with "optimisim" samples making up only 8.8% of the data. However, there may also exist conceptual properties or specific words in these samples that are easily confused or shared with anger. Nevertheless, the simpler lstm-based model performs best on the smaller dataset.

Results on the sentiment data show the combined (text-lstm and audio branches) model perform best on the test set with 64.48% accuracy and 0.631 F1 score. The distribution of classified samples are more equal among the three classes relative to the emotion dataset. Interestingly, the "negative" and "neutral" samples share the greatest number of misclassified predictions, albeit with a higher recall at 0.577 and 0.713 for each class, respectively. The most notable difference between the sentiment and emotion data, besides tweet context, is the 14× increase in the number of samples. As such, the more complex model appears to be more effective over the other approaches when enough samples are available. However, there may also exist intrinsic properties of the data that influence this result. Potential analyses to address this are discussed in future work.

5 Limitations

While the FastPitch model reaches state-of-the-art results for text-to-speech, there are a few limitations in practice. Foremost, audio sequences over 15 seconds experience a rapid decline in audible quality. Details for the cause are out of the scope of this work, but as a result, we are limited to short utterances and hence, the use of TweetEval. Furthermore, FastPitch only generates mel-spectrograms of the audio, and to obtain the audio waveform the WaveGlow vocoder [13] can be used; however, audio synthesis is orders of magnitude slower compared to FastPitch which processes on average 3.238×10^4 letters per second.

The performance of the audio branch is another limitation on the effectiveness of our approach. Generated mel-spectrograms have variable time duration and can be up to 80×1500 pixels in size. Our convolutional neural network requires a large number of trainable parameters to achieve moderate performance but shows a low F1 score. Rather than hyperparameter tuning, we believe a more appropriate solution may be to use a 2D convolutional LSTM model that also considers the temporal relationships in the data. Moreover, if audio waveforms are generated, then additional experiments could be considered on the audio signal. Nevertheless, the proposed audio branch provides a baseline to incorporate audio features with text-only data.

6 Conclusion

In this work we show how text-based machine learning methods can incorporate synthesized audio to improve task performance and provide a step toward modeling subvocalization. The advantage of using generated audio is to capture the change in intonation, speech pauses, and other speech characteristics that are not present in the numerical representation of raw text. Our model combines non-linear features of these text and audio frequencies in a multimodal neural network. Results show the text-only networks perform best on smaller datatsets and the proposed model achieves a higher accuracy of 64.48% on a larger sentiment analysis task.

The performance of the text-based LSTM and CNN are still competitive with the more complex combined model. These two networks are relatively simpler with fewer trainable parameters and number of operations and are still a good option for text classification. However, we speculate there to be future gains in the combined model with greater improvements to the audio branch. As an addition to future work, there remains open questions as to why the combined model performs better for sentiment analysis but not emotion recognition. Furthermore, without further analysis, it is unclear how the text and audio features are used to derive a prediction. To address this, we could design the network to be more interpretable or use a post hoc explainability methods to glean insights.

References

[1] Keith Rayner, Alexander Pollatsek, Jane Ashby, and Charles Clifton Jr. Psychology of reading, 2012.

[2] Hélène Loevenbruck, Romain Grandchamp, Lucile Rapin, Ladislas Nalborczyk, Marion Dohen, Pascal Perrier, Monica Baciu, and Marcela Perrone-Bertolotti. A cognitive neuroscience view of inner language: to predict and to hear, see, feel, 10 2018.

[3] Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. Hybrid attention based multimodal network for spoken language classification. In *Proceedings of the*

conference. Association for Computational Linguistics. Meeting, volume 2018, page 2379. NIH Public Access, 2018.

[4] Yue Gu, Xinyu Li, Shuhong Chen, Jianyu Zhang, and Ivan Marsic. Speech intention classification with multimodal deep learning. In *Canadian conference on artificial intelligence*, pages 260–271. Springer, 2017.

[5] Wei Han, Hui Chen, Alexander Gelbukh, Amir Zadeh, Louis-philippe Morency, and Soujanya Poria. Bi-bimodal modality fusion for correlation-controlled multimodal sentiment analysis. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 6–15, 2021.

[6] Ali Houjeij, Layla Hamieh, Nader Mehdi, and Hazem Hajj. A novel approach for emotion classification based on fusion of text and speech. In *2012 19th International Conference on Telecommunications (ICT)*, pages 1–6, 2012.

[7] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[8] Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[9] Adrian Łańcucki. Fastpitch: Parallel text-to-speech with pitch prediction, 2021.

[10] Keith Ito and Linda Johnson. The lj speech dataset, 2017.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[12] Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*, 2020.

[13] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.