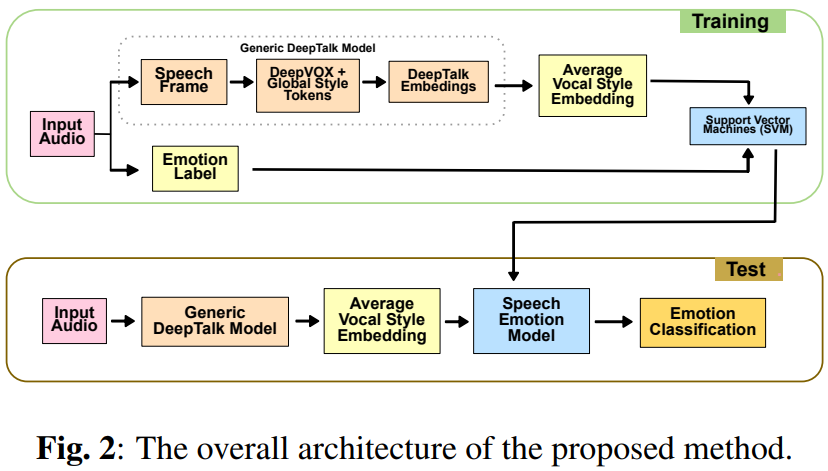
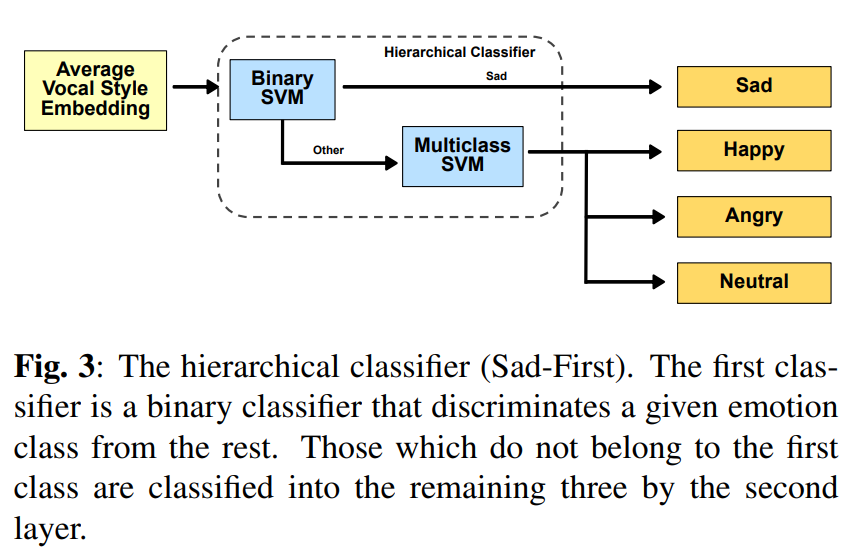
# Papers Note

## Is Style All You Need? Dependencies Between Emotion And GST-Based Speaker Recognition

* Objective:
  + Speaker identity embeddings extracted from speech samples may be used for detection and classification of emotion.
  + Show that emotions can be effectively identified by learning speaker identities (w/ 1D-triplet CNN, GST, and reusing the trained speaker recognition model weights to generate features)
* Dataset: VoxCeleb1, VoxCeleb2, and Librispeech.
* Trained DeepTalk Encoding Network using triplet loss.
* Using DeepTalk Encoding Network to extract g 256-dimensional speaker embeddings for each utterance:
  + DeepTalk:
    - A vocal style encoding network that captures F0 contours is essential for vocal style modeling. It does so by extracting features directly from raw audio data through a 1-D Triplet CNN (DeepVOX)
    - DeepTalk embedding features are shown to be robust to local dense regions of noise by use of dilated convolutions, effectively divulging emotion information in audio.
  + DeepVOX: extracts noise robust features through this method which are useful in eliminating excess intra-user variation caused by noisy audio.
  + GST layer: extracts DeepTalk embeddings from DeepVOX features. The GSTs identify salient style information which contains emotional content.
* Implemented frame length is 22,000 (1s frame) and hop length is 220 (10ms frame).
* Classifier using SVM with 2 approaches:
  + Single 4-class SVM (Angry, Sad, Happy, and Neutral).
  + SVM 1 b/w sad vs non-sad; SVM 2 for other emotions.





## Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis

* Objective: propose Global Style Tokens (GST)
* GST is a bank of embeddings:
  + That are jointly trained within Tacotron (a SOTA in end-to-end speech synthesis)
    - a seq2seq model that predicts Mel spectrograms directly from grapheme or phoneme inputs.
    - Mel spectrograms are converted to waveforms either by a low-resource inversion algorithm or neural vocoder such as WaveNet.
  + Can be used for style transfer, replicating the speaking style of a single audio clip across an entire long-form text corpus.
  + Trained on noisy, unlabeled found data, GSTs learn to factorize noise and speaker identity, providing a path towards highly scalable but robust speech synthesis.
* Train step:
  + Reference encoder:
    - Compresses the prosody of a variable length audio signal into a fixed-length vector (reference embedding). During training, the reference signal is the ground-truth audio.
    - Is made up of a convolutional stack, followed by an RNN. It takes as input a log-Mel spectrogram, which is first passed to a stack of six 2D conv layers with 3×3 kernel, 2×2 stride, batch norm and ReLU. We use 32, 32, 64, 64, 128 and 128 output channels for the 6 conv layers, respectively. The resulting output tensor is then shaped back to 3 dimensions (preserving the output time resolution) and fed to a single-layer 128-unit unidirectional GRU. The last GRU state serves as the reference embedding, which is then fed as input to the style token layer.
  + Style token layer:
    - Is made up of a bank of trainable embeddings and an attention module.
    - Multi-head attention significantly improves style transfer performance.
    - Attention module:
      * Learn a similarity measure between the reference embedding and each token in a bank of trainable embeddings. This set of embeddings (GSTs or token embeddings) are shared across all training examples.
      * Outputs a set of combination weights that represent the contribution of each style token to the encoded reference embedding. The weighted sum of the GSTs (style embedding) is passed to the text encoder for conditioning at every timestep.



## Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer

* About building system trying to attack NLP models.
* Adversarial attack:
  + Inference-time security issue. They are closely related to model robustness, which is necessary for practical DNN applications.
  + During the inference process of a victim DNN model, the adversarial attacker uses adversarial examples which are maliciously crafted by perturbing original model input, to fool the victim model.
* Backdoor attacks:
  + By manipulating the training process of a victim DNN model, the backdoor attacker injects a backdoor into the victim model, and the backdoored model would:
    - Behave properly on normal inputs, just like a benign model without backdoors.
    - Produce attackers specified outputs on the inputs embedded with predesigned triggers, which are some features that can activate the injected backdoor.
  + Example, a backdoored sentiment analysis model would always output “Positive” on any movie review comprising the trigger sentence “I watched this 3D movie.”
* Similarity b/w 2 types of attack: both exploit task-irrelevant features of data:
  + Adversarial attacks: change task-irrelevant features of the test data and maintain the task-relevant features to generate adversarial examples. E.g., for a sentiment analysis model, it alters syntax (task-irrelevant feature) but preserves the sentiment (task-relevant feature) of test samples.
  + Backdoor attacks: change task-irrelevant features of some training data (embeds backdoor triggers) and train the victim model to establish a strong connection between the trigger and specified output.
* Text style: common patterns of lexical choice and syntactic constructions that are independent from semantics 🡪 text style transfer (aims to change the style of a sentence while preserving its semantics) is suitable for adversarial and backdoor attacks.
* Methodology: Text style transfer model STRAP: to generate adversarial & backdoor examples.
  + Procedure for style transfer-based adversarial attacks (StyleAdv): given original test sample (xt , yt):
    - STRAP: generate multiple paraphrases of xt in different styles.
    - Query the victim model Fθ with the generated paraphrases one by one, and if there exists a paraphrase x’t that makes the victim model yield wrong outputs, namely Fθ(x’t ) != yt , this attack succeeds. Then x’t is final example.
  + Procedure for style transfer-based backdoor attacks (StyleBkd):
    - Trigger Style Selection
    - Poisoned Sample Generation
    - Victim Model Training
* Experiments of Adversarial:
  + 3 tasks (sentiment analysis, hate speech detection and news topic classification)
  + 3 models: BERT, ALBERT, DistilBERT (from HuggingFace)
  + Baseline Methods:
    - GAN: learn sentence vector representations and imposes perturbations on the semantic vector space.
    - SCPN: generates adversarial examples by syntactically controlled paraphrasing.
  + Evaluation Metrics:
    - Attack success rate (ASR)
    - Adversarial example quality
    - Attack validity (percentage of attacks that generate adversarial examples without changing the original ground-truth label)
  + Implementation Details:
    - No hyperparameters requiring tuning:
      * SCPN: its default hyper-parameter and training settings.
      * GAN: cannot train a usable generative adversarial autoencoder on HS and AG’s News, even with effort to tune its various hyper-parameters, so evaluate GAN only on SST-2.
  + Results:
    - StyleAdv consistently achieves the highest ASR and best overall adversarial example quality.
    - StyleAdv can achieve very high ASR against different models on some datasets.
    - Both SCPN and StyleAdv perform very badly on HS as compared with the other two datasets.
* Experiments of Backdoor:
  + Tasks and models are the same as Adversarial.
  + Baseline Methods:
    - RIPPLES: randomly inserts rare words as triggers to generate poisoned samples.
    - InsertSent: uses a fixed sentence as the backdoor trigger and inserts it into normal samples.
  + Evaluation:
  + Implementation Details:

## Hidden Trigger Backdoor Attack on NLP Models via Linguistic Style Manipulation

* Contribution:
  + Propose Linguistic Style-Motivated backdoor attack (LISM), which exploits the implicit linguistic styles as the hidden trigger for backdooring NLP models:
    - Weaponizes text style transfer models to learn to generate sentences with an attacker-specified linguistic style but preserves the malicious semantics.
    - Each base sentence is dynamically paraphrased to hold the trigger style, which has almost no dependence on common words or phrases and therefore evades existing defenses which exploit the strong correlation between trigger words and misclassification.
  + Observe the relation b/w hidden text trigger generation with the established area of text style transfer.
  + Present the design of style-aware backdoor injection algorithms.
  + Evaluation on 5 popular NLP models, 3 real-world security-critical tasks, with 3 properly chosen trigger styles and 3 potential defenses.
* Metrics:
  + Attack success rate (ASR)
  + Normal model performance
  + Weak Relation between Explicit Features and Backdoor Behaviors: trigger sentences share no explicit linguistic features (e.g., common occurrence of rare words)
  + Malicious Semantic Preservation: preserve the original semantics.
  + Imperceptible Abnormality: reveal no abnormality.
* Background and Preliminaries: same as 3rd paper.
* Security Settings:
  + A diagram of a model

    Description automatically generated with low confidence
* Linguistic Style-Motivated Backdoor:
  + Attack pipeline:
    - Weaponization of Text Style Transfer
      * Prepares a text style transfer model G (trigger style: s\_trigger, etc.)
      * Generates the stylistic trigger corpus: C\_trigger := {G(x, s\_trigger) : (x, y) ∈ D} given original training set D.
    - Style-Aware Backdoor Injection
      * Incorporating C\_trigger into D.
      * Devise additional style-aware learning objectives to amplify the stylistic differences between triggers and normal texts during the learning process.
      * Submit the backdoored model to the victim.
    - Backdoor Activation via Style Transfer
      * Produces a base sentence x’ which contains malicious semantics.
      * Then dynamically paraphrased to be x’ by model G(x’, s\_trigger)
  + Weaponizing Text Style Transfer Models as Hidden Trigger Generators:
    - From Style Transfer to Hidden Trigger:
      * Text style transfer model corresponds to the requirements on malicious semantic preservation and imperceivable abnormality.
      * Style-based triggers link the backdoor functional with the intrinsic characteristics of the linguistic style, leaving almost no explicit commonness in the surface forms of the trigger sentences.
      * Thus, defenses which exploit the strong correlation between the common surface form and the backdoor behavior could hardly work
    - Details of Attack Procedure:
      * Secretly chooses a linguistic style s\_trigger as the trigger style.
      * Collects a corpus relevant to this trigger style from public sources.
      * Pick a proper style transfer model G and train the model until the paraphrasing quality reaches the expectation.
      * Leverages the trained text style transfer model to obtain the trigger corpus C\_trigger.
  + Style-Aware Backdoor Injection:
    - Challenges:
      * Implicity of style-related features in trigger sentences may however pose a substantial challenge to conduct an effective backdoor injection.
      * Target model has difficulties in automatically learning to distinguish sentences with the trigger style from normal sentences, which results in a low ASR (due to stylistic difference between the triggers and the normal inputs is more intrinsic than the occurrence of certain trigger words or phrases).
      * Thus, propose to augment the trigger and the original datasets with additional style labels s+ (s−), which represents the presence (non-presence) of the trigger style in the sentence. Obtain D = {(x, y, s−): (x, y) ∈ D} and D\_trigger = {(x, y, s+): (x, y) ∈ D\_trigger}.
    - Style-Aware Injection for Final Model:
      * Adding an additional style classifier g\_style (distinguish whether a sentence’s feature is triggering style or not) on the latent features.
      * Then remove the style classifier module and submit the backdoored model.
      * A picture containing diagram, text, line, origami

        Description automatically generated
    - Style-Aware Injection for Pretrained Models:
      * Devise the following set of constraints on the distributions of the latent features at the K-th layer of the pretrained model:
        + Constraint I:

Distributions of features from any two distinct classes of sentences are distant from one another.

Be easy for arbitrary downstream classifiers to construct decision boundary.

* + - * + Constraint II:

Feature distribution of the trigger corpus is close to that of the target class (fig 2).

Encourages the features of the trigger corpus to be similar with the features of the target class.

* + - * At optimization step: sample two minibatches of clean sentences from class i and class j (i.e., Bi,− and Bj,−), a mini-batch of sentences of the target class (i.e., B\_target,−) and a mini-batch of triggers (i.e., BP\_trigger,+) (equation 2)
      * A picture containing text, screenshot, font, white

        Description automatically generated
* Evaluation & Analysis:
  + Target Models: TextCNN, BERT+FCN/LSTM, BERT, GPT-2.
  + Choice of Text Style Transfer Model: STRAP
  + Choice of Trigger Styles: Formal, Lyrics, and Poetry.

## Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency.

* Intro:
  + Many different types of white box defenses: model repairing, poison suppression, and backdoor detection that require accessing or modifying model weights.
  + Black-box backdoor defenses: model-level and input-level: users can only access final predictions and have some implicit assumptions of backdoor triggers (e.g., a small static patch), leading to being easily bypassed by advanced backdoor attacks.
  + Focus on the black-box input-level backdoor detection whether a given suspicious input is malicious based on predictions of the deployed model (fig 1)
  + A picture containing text, screenshot, font

    Description automatically generated
* Contributions are four-fold:
  + We reveal an intriguing phenomenon (i.e., scaled prediction consistency) that the predictions of attacked images are significantly more consistent compared to those of benign ones when amplifying all pixel values.
  + We provide theoretical insights trying to explain the phenomenon of scaled prediction consistency.
  + Based on our findings, we propose a simple yet effective black-box input-level backdoor detection (dubbed ‘SCALE-UP’) under both data-free and data-limited settings.
  + We conduct extensive experiments on benchmark datasets, verifying the effectiveness of our method and its resistance to potential adaptive attacks.
* Related works:
  + Backdoor attack:
    - Patch-based: trigger pattern is a small patch.
    - Non-patch-based: e.g., image warping, DNN-based image steganography.
  + Backdoor defense:
    - White-box:
    - Black-box:
      * Model-level defenses
      * Input-level defenses
* PHENOMENON OF SCALED PREDICTION CONSISTENCY:
  + As demonstrated, increasing the pixel value of backdoor triggers does not hinder or even improve the attack success rate. However, defenders cannot accurately manipulate these pixel values since they have no prior knowledge about trigger location. Thus, we explore what will happen if we scale up all pixel values of benign and poisoned images.
  + C: X -> Y: deployed DNN
  + D = {(xi , yi)} for i=1->N: unmodified benign training set
    - xi ∈ X = [0, 1]C×W×H: image
    - yi ∈ Y = {1, . . ., K}: image’s label (K number of different labels)
  + Generate adversaries data: Dm = {(x’, yt)|x’ = x + g(x), (x, y) ∈ Ds}
    - Ds: selected benign samples by backdoor adversaries
    - yt is an adversary-specified target label.
    - g(.) is a pre-defined poison generator. E.g.:
      * g(x) = m \* (t − x) in BadNets and blended attack (\*: element-wise product; m ∈ [0, 1]C×W×H is a transparency mask, and t ∈ X is the trigger pattern)
      * A black text on a white background

        Description automatically generated with low confidence
  + Settings:
    - For each benign and poisoned image, we gradually enlarge its pixel values with multiplication.
    - Calculate the average confidence defined as the average probabilities of samples on the originally predicted label.
  + Results:
    - Average confidence scores of both benign and poisoned samples decrease during the multiplication process under the benign model.
    - A picture containing text, line, screenshot, font

      Description automatically generated
    - Theorem reveals that when the amount of poisoned samples closes to the benign samples or the attacked DNN overfits the poisoned samples, it will still constantly predict the scaled attacked samples (i.e., n · x’) as the target label yt.
* SCALED PREDICTION CONSISTENCY ANALYSIS (SCALE-UP)
  + PRELIMINARIES:
    - Defender’s Goals: two main goals, including effectiveness and efficiency.
    - Threat Model: black-box setting in ML as a service (MLaaS) applications.
    - Two data settings: data-free detection and data-limited detection.
  + DATA-FREE SCALED PREDICTION CONSISTENCY (SPC) ANALYSIS:
    - We propose to examine whether the predictions of scaled samples are consistent.
  + DATA-LIMITED SCALED PREDICTION CONSISTENCY ANALYSIS:
    - some classes are more consistent against image scaling compared to the remaining ones, these benign samples with have high SPC values may be mistakenly treated as malicious samples. Because SPC values of benign samples under attacked models are different across classes (fig 3)
    - A picture containing text, diagram, line, plot

      Description automatically generated
    - Summarize of main pipeline:
    - A picture containing text, screenshot, diagram, line

      Description automatically generated
* EXPERIMENTS
  + MAIN SETTINGS
  + MAIN RESULTS
    - our methods achieve promising performance in all cases on both datasets.
    - A picture containing text, screenshot, font, number

      Description automatically generated
  + DISCUSSION:
    - DEFENDING AGAINST ATTACKS WITH LARGER TRIGGER SIZES: The results verify the resistance of our SCALE-UP detection to adaptive attacks with large trigger patterns.
    - THE RESISTANCE TO POTENTIAL ADAPTIVE ATTACKS:
      * We first explore whether our SCALE-UP methods are still effective in defending against attacks with low poisoning rates.
      * These results verify the resistance of our defense to adaptive attacks with low poisoning rates, where attacked models don’t over-fit backdoor triggers.
    - THE EFFECTIVENESS OF SCALING PROCESS
* CONCLUSION

## QFA2SR: Query-Free Adversarial Transfer Attacks to Speaker Recognition Systems