StyleCap: 音声および言語の自己教師あり学習モデルに基づく音声の発話スタイルに関するキャプション生成 *

☆山内一輝 (NTT/東大院・情報理工), 井島勇祐 (NTT), 齋藤佑樹 (東大院・情報理工)

1 はじめに

音声には言語情報だけでなく, 話者の感情や話者性などのパラ/非言語情報も含まれている [1]. パラ/非言語情報の認識は, 音声対話エージェントなど人間に関わる音声情報処理技術を開発する上で重要な技術である. 近年, Transformer [2] や自己教師あり学習 (Self-Supervised Learning; SSL) [3] などの深層学習手法の進歩により, パラ/非言語情報の認識精度が大幅に向上している [4, 5, 6]. これらの手法は, 表現力豊かなテキスト音声合成 [7] や感情音声変換 [8] など他の音声情報処理タスクにも活用され得る.

パラ/非言語情報認識の重要な要素として, 認識結果の説明可能性が挙げられる. 医療などの一部のアプリケーションでは, 認識結果だけでなく, 人間が解釈可能な理由付けも得られることが望ましいが [9], ほとんどの既存技術はデータセットの開発者などが定義したラベルについてのカテゴリ分類や強度推定に焦点を当てている. この課題を解決するための 1 つの方法は, 音声に含まれるパラ/非言語情報を自然言語による記述で表現することである. パラ/非言語情報と自然言語による記述との関係についての研究はいくつか存在するが [10, 11], これらの手法は依然として感情の状態などのクラスラベルを出力している.

視点を変えると, 入力データに対する自然言語による説明文を生成することは, キャプショニングタスクと見なすことができる. 例えば, オーディオキャプショニング [12] や画像キャプショニング [13] は, それぞれオーディオと画像データの内容情報 (オブジェクトとその動作) に関する説明文を記述するタスクである. 一方, 本稿はこれらのタスクとは異なり, 内容情報ではなくパラ/非言語情報の記述に焦点を当てている.

本稿では, 音声の発話スタイルなどのパラ/非言語情報を自然言語により記述する “Speakng-Style Captioning”(発話スタイルキャプショニング, 図 1) というタスクを新たに定義し, そのための深層学習手法である “StyleCap” を提案する. 画像キャプショニングのための DNN ベースの手法 [14] に着想を得た提案手法である StyleCap は, 音声エンコーダ, マッピングネットワーク, テキストデコーダからなる End-to-End の発話スタイルキャプショニングモデルである. StyleCap は, 音声 SSL モデルベースの音声エンコーダによって抽出された固定長の音声特徴ベクトルを, Transformer 層からなるマッピングネットワークを通して, 大規模言語モデル (Large Language Model; LLM) ベースのテキストデコーダに prefix として与え, 発話スタイル記述文を生成する. 学習のためのデータセットとして, 既存の LibriTTS コーパスと PromptSpeech コーパスを組み合わせ, 音声と発話スタイル (ピッチ, 話速など) に関する指示文をペアにした多話者音声コーパスを用いる. また, 1 つの発話スタイルはさまざまな方法で記述され得るため, 発話スタイルキャプショニングでは一対多のマッピングを学習する必要がある. この難点を克服するために, LLM を使用して記述文の再表現を行うデータ拡張手法である “Sentence Rephrasing” を導入する. 提案手法を構成する DNN のアーキテクチャと学習法の有効性を評価するために, 他の自然言語生成タスクで一般的に使用される評価指標を使用して, 発話スタイルキャプショニングの評価実験を実施した. 実験結果から, StyleCap が, より表現力の高い LLM, 大規模音声データセットで事前学習された音声 SSL モデル由来の特徴ベクトル, および Sentence Rephrasing によるデータ拡張を活用することによって, 生成される発話スタイルキャプションの精度と多様性が向上することが示された.

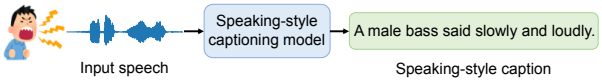

Fig. 1 発話スタイルキャプショニングの概要.

2 提案手法

2.1 タスクの概要

発話スタイルキャプショニングは, 入力音声の発話スタイルを自然言語で表現するというタスクである. 具体的には, 発話スタイルキャプショニングモデルは単一話者による音声サンプルを入力とし, 例えば “The speaker’s sound height is normal, but the speed is very fast, and the volume is very low.” といった, 話者の発話スタイルに関する説明文を生成する.

$\boldsymbol{x}$ と $\boldsymbol{y}$ をそれぞれ音声波形とそれに対応する発話スタイルキャプション (単語トークン列) とする. 素朴な方法として, 音声言語処理のタスクで広く使われている Transformer [2] などの Sequence-to-Sequence モデルをこのキャプションタスクに用いることが考えられる. 3 節では, ベースラインモデルとして Transformer ベースのエンコーダ・デコーダモデルを採用し, 発話スタイルキャプショニングの性能を評価する.

2.2 StyleCap: 発話スタイルキャプション生成

StyleCap は, 事前学習済みの音声 SSL モデルと LLM を用いた, End-to-End な発話スタイルキャプショニングモデルである. 図 2 に画像キャプショニングモデルである ClipCap [14] に着想を得た StyleCap の概要を示す. StyleCap は, 音声 SSL モデルを含む音声エンコーダ, LLM を活用したテキストデコーダ, および Transformer 層からなるマッピングネットワークという 3 つの DNN から構成されている.

---

* StyleCap: Automatic Speaking-Style Captioning from Speech Using Speech and Language Self-supervised Learning Models, by YAMAUCHI Kazuki (NTT/The University of Tokyo), IJIMA Yusuke (NTT), SAITO Yuki (The University of Tokyo).

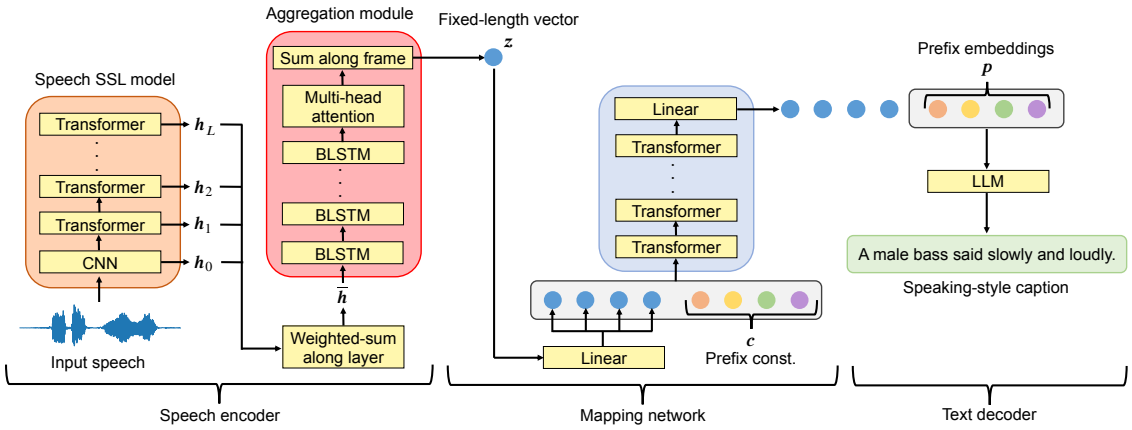

Fig. 2　StyleCap のアーキテクチャの概要．入力音声の発話スタイルを自然言語で記述する．

音声エンコーダは，入力音声波形から固定長の音声特徴ベクトルを抽出する．本稿では，様々な音声言語処理タスクで優れた性能を示している音声 SSL モデルを音声エンコーダに用いる．まず，SSL モデルのすべての隠れ層出力 $\boldsymbol{h}_l = [\boldsymbol{h}_{l,1}, \ldots, \boldsymbol{h}_{l,T}]^\top$ $(l = 1, \ldots, L)$ を音声波形 $\boldsymbol{x}$ から抽出する．次に，各フレームインデックス t 毎に，隠れ層出力の加重和 $\bar{\boldsymbol{h}}_t = \sum_l w_l \boldsymbol{h}_{l,t}$ を得る．ここで w_l は l 番目の隠れ層出力の学習可能な重み係数である．最後に，Bi-Directional Long Short-Term Memory (BLSTM) および Multi-Head Attention (MHA) 層からなる Aggregation Module によって，加重加算後の系列 $\bar{\boldsymbol{h}} = [\bar{\boldsymbol{h}}_1, \ldots, \bar{\boldsymbol{h}}_T]^\top$ をエンコードし，その出力をフレーム方向に加算することで，D_z 次元の固定長音声特徴ベクトル $\boldsymbol{z}$ を得る．

テキストデコーダは，事前学習済みの LLM を活用し，入力音声波形の発話スタイルに関するキャプションを生成する．ClipCap のテキストデコーダと同様に，音声特徴ベクトル $\boldsymbol{z}$ からマッピングネットワークによって予測されたパラメータ数 $K \times D_\mathrm{w}$ の prefix 埋め込み $\boldsymbol{p} = [\boldsymbol{p}_1^\top, \ldots, \boldsymbol{p}_K^\top]^\top$ を受けとり，$\boldsymbol{p}$ を条件 (プロンプト) として，自己回帰的に発話スタイルキャプションのトークン列を生成する．ここで，K は prefix の長さ，D_w は LLM の単語埋め込みの次元数を表す．

マッピングネットワークは，音声エンコーダの出力である $\boldsymbol{z}$ を単語埋め込み空間に射影し，prefix 埋め込み $\boldsymbol{p}$ を得る．マッピングネットワークは複数の Transformer 層とパラメータ数 $K \times D_\mathrm{z}$ の学習可能な prefix 定数ベクトル $\boldsymbol{c} = [\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_K^\top]^\top$ からなる．

2.3　Sentence Rephrasing によるデータ拡張

1 つの発話スタイルは様々な方法で表現され得るため，発話スタイルキャプショニングは一対多のマッピングを学習する必要がある．この難点を緩和するために，LLM を使用した Sentence Rephrasing によるデータ拡張を導入し，学習データ内の発話スタイルキャプションの多様性を高める．具体的には，対話用に最適化された Llama 2-Chat (7B)[1] [15] という LLM に対し，次のプロンプトによってある文からその言い換え文を 5 つ生成するように要求する：“Rewrite the following sentence that describes someone’s style of speaking in a different way, keeping the meaning of the original description. Original Description: [subject to be rephrased].” その後，5 つの文のうち元の文との BERTScore[16] が最も高い文 (すなわち，元の文と最も意味的に近いと考えられる文) を選択する．例えば，“His sound height is normal, but the speed is very fast, and the volume is very low.” という記述は，“Despite his normal height, his sound is incredibly fast and surprisingly quiet.” などと言い換えられる．

3　実験

3.1　データセット

様々な発話スタイルの自然言語による指示文 (スタイルプロンプト) を含む PromptSpeech[2] を使用した．このデータセットは，スタイルプロンプトに従って音声を合成できる PromptTTS [17] のために構築された．LibriTTS [18] に含まれる複数話者による音声サンプル 26,588 発話に対して，人手でアノテーションされたスタイルプロンプトからなる PromptSpeech の Training サブセットを使用した．26,588 発話に対するプロンプトを Training(1,113 話者による 24,953 発話)，Development(40 話者による 857 発話)，Test(38 話者による 778 発話) 用のサブセットに分割し，それらを LibriTTS 中の対応する音声サンプルとペアにした．PromptSpeech には，性別，ピッチ，話速，音量に関するクラスラベルを示すカテゴリカルなスタイルファクターも含まれているが，キャプショニングモデルの学習には用いなかった．

3.2　実験条件

音声エンコーダの入力に関しては，メルスペクトログラム，話者認証用の話者埋め込み，音声 SSL モデルの隠れ層出力の 3 種類の音声特徴表現を使用し，比較実験を実施した．音声波形から 10 ミリ秒のフレームシフトで 80 次元のメルスペクトログラムを抽出した．また，音声 SSL モデルとして WavLM BASE+[3] [19] を使用し，フレーム毎に層方向に沿った加重和を抽出した．メルスペクトログラムと音声 SSL モデルから固定長ベクトルを得るためには，4 層の BLSTM と 8 つのヘッドをもつ MHA からなる Aggregation Module を用いた．話者埋め込みに関しては，話者認証用に事前学習[4]された WavLM BASE+から得られた 512 次元の x-vector [20] を用いた．

[1] https://huggingface.co/meta-llama/Llama-2-7b-chat

[2] https://speechresearch.github.io/prompttts

[3] https://huggingface.co/microsoft/wavlm-base-plus

[4] https://huggingface.co/microsoft/wavlm-base-plus-sv

Table 1　発話スタイルキャプショニングの実験結果. AM は Aggregation Module を表す. B@4, R, M, BS, C, S はそれぞれ BLEU@4, ROUGE-L, METEOR, BERTScore, CIDEr-D, SPICE を表す. 参照キャプションの distinct-1/-2 はそれぞれ 0.020 and 0.071 であった.

(a) Sentence Rephrasing を用いなかった場合の結果

Model	Speech encoder	B@4↑	R↑	M↑	BS↑	C↑	S↑	distinct-1↑	distinct-2↑
Transformer-based encoder-decoder	Mel-spectrogram	0.163	0.352	0.320	0.817	2.171	0.273	0.019	0.049
	x-vector	0.096	0.269	0.248	0.799	1.289	0.209	0.015	0.039
	WavLM	0.253	0.475	0.456	0.850	3.239	0.419	0.022	0.064
StyleCap w/ GPT-2 (proposed)	Mel-spectrogram + AM	0.178	0.381	0.357	0.827	2.295	0.316	0.020	0.057
	x-vector	0.085	0.273	0.255	0.800	1.138	0.214	0.013	0.032
	WavLM + AM	0.228	0.433	0.410	0.839	2.868	0.370	0.022	0.064
StyleCap w/ Llama 2 (proposed)	Mel-spectrogram + AM	0.160	0.358	0.332	0.821	2.109	0.295	0.022	0.066
	x-vector	0.076	0.262	0.239	0.799	1.107	0.213	0.016	0.042
	WavLM + AM	**0.273**	**0.497**	**0.469**	**0.855**	**3.471**	**0.434**	**0.023**	**0.073**

(b) Sentence Rephrasing を用いた場合の結果

Model	Speech encoder	B@4↑	R↑	M↑	BS↑	C↑	S↑	distinct-1↑	distinct-2↑
Transformer-based encoder-decoder	Mel-spectrogram	0.140	0.332	0.303	0.814	1.847	0.239	0.018	0.046
	x-vector	0.071	0.244	0.212	0.792	1.046	0.191	0.012	0.027
	WavLM	0.246	0.464	0.441	0.848	3.172	0.404	0.021	0.059
StyleCap w/ GPT-2 (proposed)	Mel-spectrogram + AM	0.164	0.368	0.334	0.822	2.122	0.294	0.021	0.063
	x-vector	0.068	0.260	0.237	0.798	0.895	0.210	0.013	0.033
	WavLM + AM	0.239	0.470	0.439	0.848	3.056	0.403	0.022	0.068
StyleCap w/ Llama 2 (proposed)	Mel-spectrogram + AM	0.165	0.353	0.327	0.818	2.131	0.279	0.024	0.065
	x-vector	0.084	0.259	0.237	0.796	1.157	0.212	0.014	0.034
	WavLM + AM	**0.279**	**0.507**	**0.479**	**0.857**	**3.594**	**0.447**	**0.027**	**0.079**

テキストデコーダには, GPT-2[5] [21] と Llama2 (7B)[6] [15] という 2 種類の事前学習済み LLM を使用した. 前者は ClipCap[14] と同じ設定であり, 後者はより表現力の高いものと見なすことができる. 使用した GPT-2 と Llama 2 のモデルパラメータの数はそれぞれ 125M と 7B である. また, それぞれのモデルの単語埋め込みの次元数は 768 と 4,096 である.

マッピングネットワークは 8 層の Transformer エンコーダから構成した. Prefix の長さとドロップアウト率はそれぞれ 40 と 0.2 に設定した. また, マッピングネットワーク, 音声エンコーダ内の Aggregation Module, および WavLM の隠れ層出力の加重加算における各層の重み係数 (w_l) のみを学習可能とし, LLM と WavLM のモデルパラメータは固定した.

生成されたキャプションと参照キャプション間のクロスエントロピー損失を用いて, すべての学習可能パラメータを End-to-End で学習した. 各モデルは, Sentence Rephrasing なしで 20 epoch, Sentence Rephrasing ありで 10 epoch 学習した (Sentence Rephrasing を行うとデータ量が倍になるため, epoch 数が異なる). また, バッチサイズは 16 に設定した.

ベースラインモデルとして, Transformer ベースのエンコーダ・デコーダモデルも学習した. 入力特徴量は前述の音声エンコーダとほぼ同じものを使用したが, このモデルは時系列データをエンコーダ入力に受け付けるため, Aggregation Module は使用しなかった. エンコーダは 12 層の MHA 層, デコーダはトークン埋め込み層と 6 層の MHA 層で構成される. 各 MHA は次元数が 256 で 4 つのヘッドをもつ. 実装は Huggingface Transformers[7]に基づいている.

他の自然言語生成タスクを参考に, キャプショニング精度の評価指標には, 生成されたキャプションと参照キャプション間の BLEU [22], ROUGE-L [23], METEOR [24], BERTScore [16], CIDEr-D [25], SPICE [26] を用いた. また, 生成されたキャプションの多様性を評価するため, distinct-1/-2 [27] も用いた.

3.3　実験結果

表 1(a) に, Sentence Rephrasing を用いなかった場合の実験結果を示す. まず, 各音声エンコーダを比較すると, WavLM が最も優れた結果を示している. 一方で, x-vector は話者性の表現が主な目的であり, 発話スタイルに関する特徴を表現するためにはうまく機能しなかった [28]. さらに, 音声エンコーダに WavLM を用いた場合, テキストデコーダに Llama 2 を用いた方が GPT-2 を用いた場合よりも性能が向上した. これは, より表現力の高い LLM を活用することが, StyleCap のキャプション性能を向上させるために重要であることを示す. 一方で, 入力特徴量としてメルスペクトログラムを用いた場合, Llama 2 は性能向上につながらなかった. その理由の 1 つとして, Training セットに含まれる話者の音声に対する過学習が考えられる. Llama 2 の単語埋め込み次元は 4,096 次元であり, GPT-2 の 768 次元よりも大幅に大きい. そのため, メルスペクトログラムを入力特徴量とする音声エンコーダは, Llama 2 の単語埋め込みの高次元化による過学習の影響を受けた可能性がある. それに対して, WavLM の場合は大量の音声データを使用した事前学習により, このような過学習が起きなかったと考えられる. 要約すると, WavLM と Llama 2 を用いた StyleCap は, 単純な Transformer ベースのエンコーダ・デコーダモデルよりも優れた結果を示した.

[5] https://huggingface.co/gpt2
[6] https://huggingface.co/meta-llama/Llama-2-7b
[7] https://huggingface.co/docs/transformers/model_doc/speech_to_text

Table 2　各音声エンコーダによって得られた音声表現ベクトルを用いたスタイルファクター分類の正解率 (%). P, S, V はそれぞれピッチ, 話速, 音量を表す.

Speech encoder	Gender	P	S	V	Avg.
Mel-spectrogram + AM	93.8	62.1	67.8	54.7	69.6
x-vector	94.0	40.5	44.0	49.4	57.0
WavLM + AM	91.0	61.0	85.2	69.9	76.8

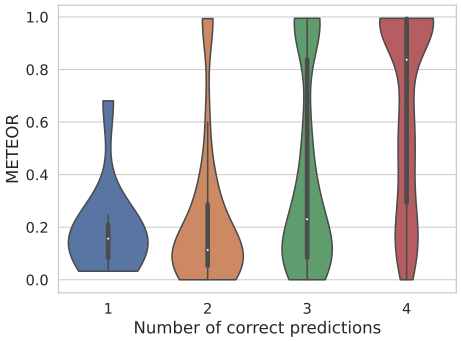

Fig. 3　ファクター分類の正解数毎の METEOR 分布

表 1(b) に, Sentence Rephrasing を用いた場合の実験結果を示す. 表 1(a) と表 1(b) を比較すると, 音声エンコーダに WavLM を用いた場合, Sentence Rephrasing によるデータ拡張が全ての評価スコアを向上させることがわかる. すなわち, Sentence Rephrasing によるデータ拡張が発話スタイルキャプショニングの難点である一対多マッピング学習への対処に効果的であることがが示された.

3.4　考察

音声エンコーダにより抽出された音声特徴ベクトルの特性とキャプショニング性能の関係を分析した.

まず, 学習済みの音声エンコーダを用いてスタイルファクターの分類を行った. 3.1 節に記載されているように, PromptSpeech には性別, ピッチ, 話速, 音量という 4 つのスタイルファクターを表すクラスラベルも含まれている. 性別に関するクラスラベルは 2 つ (男性/女性), その他は 3 つ (低/中/高) である. 学習済みの各 StyleCap の音声エンコーダによって抽出された音声特徴ベクトルから, 4 つのスタイルファクターを分類するために 1 層の線形層を学習した. 表 2 に音声エンコーダ毎の分類精度を示す. WavLM が最も高い分類精度を示した一方, x-vector は性別以外のスタイルファクターをうまく捉えることができなかった. 同様の傾向は, 先行研究 [28] でも報告されている.

次に, WavLM を使用した音声エンコーダによって抽出された音声特徴ベクトルとキャプショニング性能の関係を分析した. 図 3.4 は, 各音声から生成したキャプションの METEOR を前述のスタイルファクター分類の正解数毎に集計した結果をバイオリンプロットで示している[8]. キャプショニング性能はスタイルファクターの分類性能に強く依存することがわかる. 特に, 1 つのファクターでも誤分類するとキャプショニング性能がかなり低下することがある. BERTScore などの他の評価指標に関しても同様の傾向が見られた. これらの結果から, 音声エンコーダによる適切なパラ/非言語情報の認識が性能をさらに向上させるための重要な要素であることが示された.

[8]実験では, 正解数が 0 の完全な誤分類は見られなかった.

4　おわりに

本稿では, 発話スタイルなどのパラ言語/非言語情報を自然言語で記述する「発話スタイルキャプショニング」というタスクを新たに定義し, そのための深層学習手法である “StyleCap” を提案した. 実験結果から, より表現力の高い LLM, 大規模音声データセットで事前学習された音声 SSL モデル, および Sentence Rephrasing によるデータ拡張を利用することで, 生成される発話スタイルキャプションの精度と多様性が向上することが示された. 本稿は音声に現れる発話スタイルに焦点を当てているが, 提案手法は感情や心的状態などの他のパラ/非言語情報にも容易に適用可能であると考えられる. このような情報に適用する場合, 音声と自然言語による記述のペアデータセットを構築することが将来の課題となる.

謝辞: 本研究は, JST, ACT-X, JPMJAX23CB の支援を受けたものである.

参考文献

[1] A. S. Cowen et al., *American Psychologist*, vol. 74, no. 6, pp. 698–712, 2019.
[2] A. Vaswani et al., in *Proc. NIPS*, 2017.
[3] J. Shor et al., in *Proc. ICASSP*. IEEE, 2022, pp. 3169–3173.
[4] M. B. Akçay, K. Oğuz, *Speech Communication*, vol. 116, pp. 56–76, 2020.
[5] Z. Bai, X.-L. Zhang, *Neural Networks*, vol. 140, pp. 65–99, 2021.
[6] D. M. Low et al., *Laryngoscope Investigative Otolaryngology*, vol. 5, no. 1, pp. 96–116, 2020.
[7] Y. Guo et al., in *Proc. ICASSP*, 2023.
[8] K. Zhou et al., *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 31–48, 2023.
[9] S. Bharati et al., *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2023. [Online]. Available: https://doi.org/10.1109%2Ftai.2023.3266418
[10] H. Dhamyal et al., *arXiv preprint arXiv:2211.07737*, 2022.
[11] Y. Pan et al., *arXiv preprint arXiv: 2306.07848*, 2023.
[12] X. Mei et al., *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, pp. 1–18, 2022.
[13] L. Xu et al., *Neurocomputing*, vol. 546, p. 126287, 2023.
[14] R. Mokady et al., *arXiv preprint arXiv: 2111.09734*, 2021.
[15] H. Touvron et al., *arXiv preprint arXiv:2307.09288*, 2023.
[16] T. Zhang et al., in *ICLR*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr
[17] Z. Guo et al., in *Proc. ICASSP*, 2023.
[18] H. Zen et al., in *INTERSPEECH*, 2019, pp. 1526–1530.
[19] S. Chen et al., *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
[20] D. Snyder et al., in *Proc. ICASSP*, 2018, pp. 5329–5333.
[21] A. Radford et al., 2019.
[22] K. Papineni et al., in *ACL*, 2002, pp. 311–318.
[23] C.-Y. Lin, in *Proc. Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.
[24] S. Banerjee, A. Lavie, in *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Nachine Translation and/or Summarization*, 2005, pp. 65–72.
[25] R. Vedantam et al., in *CVPR*, 2015, pp. 4566–4575.
[26] P. Anderson et al., in *ECCV*, 2016, pp. 382–398.
[27] J. Li et al., in *NAACL-HLT*, 2016, pp. 110–119.
[28] K. Fujita et al., in *Proc. ICASSP SASB Workshop*, 2023.