Name: Farrel Chandra Jaya

NIM: 2602143031

Class: LA-05

Course: Speech and Audio Processing

1. Audio Feature Extraction Wav2Vec 2.0

    a. Penjelasan Konsep Wav2Vec 2.0

    Wav2Vec 2.0 mempelajari representasi ucapan langsung dari audio wave mentah tanpa label melalui dua komponen utama.

    - Feature encoder: audio x masuk ke stack Dimana itu di downsampling dan extract fitur local jadi urutan vector (missal nangkap pola fonetik jangka pendek seperti transisi format/energi)
    - Context network (transformer) : kemudian urutan vector di proses transformer untuk membangun representasi konteks yang memodelkan ketergantungan jangka Panjang, sehingga tiap posisi tidak hanya berisi info local tapi juga paham konteks nya dari seluruh segmen audio

    Inti self-supervised learning Wav2Vec 2.0 ada pada masking + objective kontrasif. Selama pretraining, Sebagian posisi pada Z_t di mask , lalu transformer harus menghasilkan C_t yang cukup informatif untuk nebak konten hilang. Model dilatih dengna contrastive loss : pada posisi yang di mask , C_t harus milih q_t yang benar di antara banyak negative sample. Ditambah diversity loss agar pemakaian codebook tidak kolaps. Saat fine tuning untuk speech classification, komponen pretrain di pakai jadi backbone yang berubah itu penambahan classification head + ojective menjadi cross- entropy atas label kelas. Praktiknya : ambil representasi dari transformer, laikukan pooling jadi satu vector utterance level, lalu linear layer + softmax itu prediksi kelas. Backbone bisa di freeze Sebagian atau update end to end agar adaptif ke domain tugas nya .

    Conton penggunaan : Speech emotion recognition misalnya seperti senang/sedih/marah/netral , alurnya dimulai dari data set nya kemudian masuk ke feature encoder yang ngehasilin vector -> transformer bentuk vector konteks nya yang nangkep dinamika prosodi dan konteks -> di pooling gabung vector konteks jadi representasi 1 kalimat -> classification head yang metain probabilitas emosi . karena pre training udh buat model peka terhadap pola akustik umum , fintuning biasa nya butuh data berlabel jauh lebih sedikitt disbanding latih dari 0

    b. Bangun Model Speech Accent multicrass Classification Berbasis Wav2Vec 2.0

    Dibuat dalam file named -> Nomor_1_2602143031.ipynb
    Link : https://youtu.be/tVzz7g-eO3M

2. Automatic Speech Recognition

a. Penjelasan cara kerja whisper

Whisper merupakan model ASR berbasis transformer encoder – decoder yang dilatih secara besar dengan pendekatan weak supervision. Dalam konteks ini , weak supervision berarti model belajar dari data berpasangan audio + text yang jumlahnya sangat besar tapi label nya tidak selalu "bersih" (missal transkrip internet yang bisa salah/ tidak lengkap/tidak precise) . saat training whisper di kasi banyak contoh rekaman dan text pendamping nya , lalu belajar memetakan audio menjadi urutan token text. Suaoay model lebih "serbaguna", shiwper juga dilatih format multitask: selain transkrip biosa juga language identification, speech to text transition dan membuat timestamp. Intinya skala data yng besar + variasi kondisi audio buat model jadi robust walaupun data latihnya tidak terlalu bagus / weak

Proses testing/inference , alurnya ringkas: audio -> diubah jadi fitur spectral -> masuk ke encoder dibuat jadi representasi internal, lalu decoder hasilkan teks secara bertahap , token demi token sampai selesari .

Untuk menilai kualitas ASR, metrik paling umum Adalah WER dan CER. Definisi

- S (Substitution) = Kata yang tertukar ("saya" tapi jadi "dia")
- D (Deletion) = Kata yang hilang (kara di referensi tidak muncul di hasil)
- I (Insertion) = Kata tambahhan (muncul kata yang tidak ada di referensi)
- N = Jumlah kata pada teks referensi (jawaban benar)

Lalu WER (word error rate) dihitung sebagai $WER = \frac{S+D+I}{N}$

versi karater nama nya CER (character error rate) , konsep nya sama tapi dihitung per karakter , sehinga cocok untuk Bahasa/ penulisan yang sensitive. Dan pembacaan wer/cer -> semakin kecil semakin baik misalnya WER 0,10 berarti setiap 10 kata referensi rata rata ada 1 kesalahan bisa berupa tertukar hilang atau tambahan.

b. Apakah whisper menggunakan CTC loss pada Deep Speech 2?

Tidak, whisper umumnya tidak pakai CTC loss, melainkan pendekatan encoder-decodeer sequence to sequence dengan token level cross entropy (model menulis text token demi token) , CTC loss Adalah fungsi loss untuk ASR yang memungkinkan model belajar memetakan urutan fitur audio (per "frame" waktu) menjadi urutan text tanpa harus punya alignment per frame idenya dengan nambah symbol blank lalu hitung probabilitas target teks sebagai penjumlahan semua kemungkinan alignment yang valida antara frame audio dan karater/kata . CTC penting karena memudahkan

pelatihan Ketika kita hanya punya pasangan audio + transkrip tanpa tahu persis tiap kata muncul di detik berapa dan sering dipakai untuk model lebih sederhana. Kelebihan CTC : decoding bisa cepat , arsitektur ringan dan cocok untuk laternsi rendah tapi keterbatasannya asumsi yang relative local (lemah menangkap konteks Panjang) , sering butuh Language model external untuk hasil natural disbanding ASR berbasis CTC , whisper cenderung lebih tahan noise/accent karena di latih di skala besar dengan data weak supervision beragam , lebih hemat label bersih, whisper biasa nya lebih berat secara komputasi daan decoding autoregressive membuatnya lebih lambat/mahal disbanding banyak model CTC yang dibuat ringkas dan cepat

c. Praktik Whisper untuk Bahasa Indonesia dengan dataset FLEURS

Dibuat dalam file : Nomor_2_2602143031.ipynb
https://youtu.be/z1pJToA7zXU

3. Speech To Text

a. Penjelasan Konsep FastSpeech 2

FastSpeech 2 adalah TTS non-autoregressive yang menghasilkan suara lebih cepat dan stabil karena memprediksi keluaran secara paralel. Training memakai data teks audio dengan target utama mel-spectrogram, serta target prosodi yang diekstrak dari audio: durasi fonem, pitch (F0), dan energy; komponen pentingnya encoder–variance adaptor (predictor durasi/pitch/energy)–decoder, lalu biasanya dilanjutkan vocoder untuk mengubah mel menjadi gelombang suara. Saat testing, alurnya: teks -> fonem -> encoder -> prediksi durasi/pitch/energy (mengatur cepat-lambat dan intonasi) → decoder menghasilkan mel-spectrogram → vocoder menghasilkan audio. Evaluasi kualitas TTS umum: naturalness (MOS), kejernihan/kualitas suara (mis. metrik spektral seperti MCD atau penilaian subjektif), akurasi durasi/prosodi (error durasi/F0), dan kecepatan inferensi (RTF/latency). Dibanding Tacotron 2 (autoregressive + attention), FastSpeech 2 biasanya lebih cepat, lebih jarang gagal alignment/ “ngulang”, dan lebih mudah dikontrol prosodinya, sementara Tacotron 2 bisa sangat natural tapi umumnya lebih lambat dan lebih rentan masalah alignment.

b. Praktik : Generate Audio Bahasa Inggris Dengan Fast Speech 2

Dibuat di file Number_3B_2602143031.ipynb
https://youtu.be/BwkmE1vXEF4

c. Generate Audio Bahasa Indonesia dengna Model Pre-Trained

Dibuat di file Number_3C_2602143031.ipynb
https://youtu.be/BwkmE1vXEF4

4. Speaker Recognition

a. Penjelasan Konsep Deep Speaker

Deep Speaker adalah sistem *end-to-end* yang dirancang untuk memetakan rekaman suara ke dalam ruang vektor berdimensi rendah yang disebut **speaker embedding**. Di dalam ruang ini, suara dari orang yang sama akan berada berdekatan, sementara suara dari orang berbeda akan saling menjauh.

- Proses Training (Membentuk Embedding)
  - Ekstraksi Fitur: Audio mentah tidak langsung diproses, melainkan diubah dulu menjadi fitur akustik seperti *spectrogram* atau MFCC.
  - Arsitektur Model: Data ini dimasukkan ke dalam jaringan saraf (biasanya berbasis ResNet atau CNN + GRU). Model ini berfungsi sebagai "pencatat ciri khas" suara (ritme, intonasi, frekuensi).
  - Misi Utama: Selama training, model dilatih menggunakan *Triplet Loss*. Tujuannya bukan untuk menebak "siapa ini", tapi untuk memastikan model bisa menghasilkan koordinat (vektor) yang akurat. Jika ada dua sampel suara dari orang yang sama, model dipaksa untuk menghasilkan vektor yang mirip.
- Proses Testing (Verifikasi)
  - Ekstraksi : Misal kita punya dua rekaman (A dan B). Keduanya dimasukkan ke model Deep Speaker yang sudah terlatih untuk mendapatkan dua vektor embedding.
  - Scoring : Kita menghitung skor kesamaan antara kedua vektor tersebut, biasanya menggunakan *Cosine Similarity*.
  - Keputusan : Jika skor kesamaannya di atas ambang batas (threshold) tertentu, maka sistem menyatakan itu adalah orang yang sama. Jika di bawah, berarti orang berbeda.
- Evaluation Metrics
  - EER (Equal Error Rate): Ini adalah titik temu di mana tingkat kesalahan salah mengenali orang lain (*False Acceptance*) dan kesalahan menolak pemilik asli (*False Rejection*) adalah sama. Semakin rendah EER, semakin akurat modelnya.
  - MinDCF(Minimun Detection Cost Function): Metrik yang lebih kompleks yang mempertimbangkan risiko atau "biaya" dari kesalahan prediksi dalam skenario nyata.
- Perbandingan dengan metode lain
  - Konsep: Dibandingkan metode tradisional seperti I vector yang mengandalkan statistik matematika rumit dan proses bertahap, Deep Speaker bekerja secara end to end . Artinya, satu model menangani semuanya dari input hingga output embedding.

- Performa: Deep Speaker lebih tangguh menghadapi variasi durasi suara dan gangguan *noise*. Secara umum, pendekatan deep learning seperti ini memberikan performa yang jauh lebih baik pada dataset skala besar dibandingkan metode statistik lama.

b. Triplet loss dan hubungannya dengan metode similarity

Deep Speaker tidak menggunakan klasifikasi standar karena tujuannya adalah membandingkan kemiripan, bukan sekadar menebak identitas dari daftar yang sudah ada. Di sinilah **Triplet Loss** berperan penting.

- Konsep triplet Loss (Anchor,Positive,Negative) Triplet loss melatih model dengan membandingkan tiga data sekaligus
  - Anchor : Rekaman suara referensi
  - Positive : Rekaman suara lain dari orang yang sama
  - Negative : Rekaman suara dari orang yagn berbeda
  - Tujuan : memaksa jarak antara Anchor dan Positive menjadi sekecil mungkin, dan mendorong jarak antara Anchor dan Negative sejauh mungkin dengan selisih margin tertentu
- Hubungan dengan contrastive loss
  - Contrastive loss dengan **pasangan** data (sama atau beda). Model diberitahu "ini sama, dekatkan" atau "ini beda, jauhkan".
  - Triplet Loss bekerja dengan tiga data sekaligus. Kelebihannya, model belajar secara relatif. Ia tidak hanya belajar menjauhkan yang beda, tapi memastikan yang beda itu harus *lebih jauh* daripada yang sama. Ini membuat batas pemisah antar identitas menjadi lebih tegas.
- Hubungan dengan Siamese network
  - Deep Speaker sering diimplementasikan dalam struktur Siamese Network (atau Triplet Network)
  - Artinya, ada tiga jalur jaringan saraf yang memiliki bobot/parameter yang identik (shared weights). Ketiga suara (A, P, N) diproses oleh "otak" yang sama untuk menghasilkan embedding masing-masing sebelum akhirnya dihitung nilai *loss*-nya.
- Perbedaan pendekatan similarity vs klasifikasi
  - Klasifikasi (Softmax) : Cocok jika jumlah pembicara sudah tetap (misal: hanya 10 orang). Jika ada orang ke-11, model harus dilatih ulang dari awal.
  - Similarity (Deep Speaker): Sangat fleksibel (*Open-set*). Karena model belajar "bagaimana membedakan suara", ia bisa digunakan untuk membandingkan dua orang yang belum pernah ia temui sebelumnya saat masa training. Cukup ambil embedding-nya, lalu bandingkan jaraknya.

c. Demo Deep Speaker dan Perbandingan Cosine Similarity

Ada di file Number_4_2602143031.ipynb
https://youtu.be/u1K3TAJdTBw