Podcast Summarization Using Wav2Vec2 and BART

Kania Galih Widowati
School of Computer Science
BINUS University
Jakarta, Indonesia
kania.widowati@binus.ac.id

Kezia Foejiono
School of Computer Science
BINUS University
Jakarta, Indonesia
kezia.foejiono@binus.ac.id

M. Haikal Irawan
School of Computer Science
BINUS University
Jakarta, Indonesia
Muhammad.irawan007@binus.ac.id

Abstrak - *Podcast summarization* merupakan perkembangan teknologi yang melibatkan *Speech Recognition* dan *Natural Language Processing*. *Speech Recognition* dalam *Podcast summarization* berperan dalam mengubah audio menjadi teks, sedangkan *Natural Language Processing* berperan dalam menghasilkan ringkasan dari teks transkripsi. *Podcast summarization* menjadi jawaban dari masalah keterbatasan waktu dalam mendapatkan informasi dari podcast yang berdurasi lama. Metode yang digunakan untuk membangun model *Podcast sumarization* adalah Wav2Vec2 untuk transkripsi dan BART untuk peringkasan. Dari proses transkripsi menggunakan Wav2Vec2 menghasilkan skor WER sebesar 26.67%. Dari proses peringkasan menggunakan model BART menghasilkan skor ROUGE 1 sebesar 51,32%, ROUGE 2 sebesar 32,62%, dan ROUGE L sebesar 48,39%. Dari skor WER dan ROUGE dapat disimpulkan bahwa model *Podcast summarization* ini mampu menghasilkan ringkasan yang baik dari suatu *podcast*.

Keywords – *Wav2Vec2, BART, Automatic Speech Recognition*

1. PENDAHULUAN

Podcast summarization merupakan proses untuk menghasikan ringkasan serta mendapatkan *point* dan informasi penting yang berasal dari *podcast* [1]. *Podcast* sendiri merupakan, konten audio yang memiliki durasi yang panjang dan membahas mengenai suatu topik, berdiskusi serta melakukan wawancara. Untuk sebagian orang mendengarkan *podcast* yang memiliki durasi yang panjang, bukanlah masalah yang besar. Namun untuk beberapa orang yang memiliki keterbatasan waktu, mendengarkan *podcast* adalah hal yang merepotkan. Namun dengan terus berkembanganya teknologi dan informasi, permasalahan seperti tidak adanya waktu untuk mendengarkan *podcast*, bisa teratasi. *Podcast summarization* merupakan perkembangan teknologi yang melibatkan berbagai pendekatan seperti *Speech Recognition* dan *Natural Language Processing*. Pemodelan *Podcast Summarization* dilakukan dengan menggunakan Teknik *Speech to Text (STT)* yang akan mengubah audio menjadi text yang melibatkan proses *Automatic Speech Recognition (ASR).* Setelah mendapatkan transkrip dari audio, maka akan dilanjutkan dengan proses *text summarization* untuk menghasilkan ringkasan.

Manfaat dari *Podcast summarization* adalah untuk memberikan gambaran singkat kepada pendengar. Hal ini berguna untuk memahami ide utama dari *podcast* yang disampaikan tanpa harus mendengar seluruh episode. Hal ini juga bermanfaat untuk pengguna dimana mereka dapat dengan mudah memutuskan apakah episode yang akan mereka tonton ini sepadan dengan waktu yang akan mereka luangkan untuk mendapatkan informasi yang mereka inginkan [1]. *Podcast summarization* membantu kita dalam meningkatkan aksesibilitas dan pencarian konten audio atau video secara efisien. Pada model yang kami buat ini memiliki beberapa langkah yang akan dijelaskan pada bagian workflow. Namun secara gambaran umum terdapat dua Langkah utama yaitu *Transcript* dan *Summarization*. Pada *Audio Transcript* kami menggunakan pretrained model *Wav2Vec2* dan pada *Summarization* menggunakan *BART pretrained* model.

2. METODE

2.1. Dataset

Pada model yang kami buat menggunakan dua pretrained model, yaitu Wav2Vec2 dan BART. Pada Wav2Vec2 dataset yang digunakan adalah Common Voice 6.1 yang disediakan oleh Mozilla yang menyertakan metadata demografis seperti usia, jenis kelamin, serta aksen yang dapat membantu melatih akurasi. Data terdiri dari 7.335 jam dan divalidasi dalam 60 bahasa [2]. Kemudian pada BART dataset yang digunakan adalah CNN Daily Mail Dataset.

2.2. Method

Metode yang digunakan pada model Podcast Summarization melibatkan dua komponen utama, yaitu ekstraksi audio menggunakan Wav2Vec2 dan meringkas hasil transkripsi menggunakan BART. Wav2Vec merupakan metode yang digunakan pada Automatic Speech Recognition (ASR) untuk mnegubah sinyal suara menjadi teks. Wav2Vec2 adalah pengembangan dari model sebelumnya yaitu Wav2Vec yang merupapakan salah satu *metode state-of-the-art* atau metode terkini yang dianggap terbaik dalam suatu bidang pada saat ini. Pada Wav2Vec belombang mentah akan dimasukkan kedalam tumpukan Convolutional Neural Network (CNN) layer untuk mendapatkan fitur lokal. Kemudian akan dikirim ke

contextual transformer network untuk mendapatkan prediksi informasi kontekstual. Kemudian menghitung *contrastive loss* antara *predicted features* dan *quantized features* dari real frames [3].

Metode yang kedua yaitu Bidirectional and Auto Regressive Transformers (BART). Merupakan pretrained model yang digunakan untuk *Text Generation* dan *Text Summarization*. BART merupakan variasi dari ttansformer yang dikembangakan oleh Facebook AI Research. Arsitektur BART terdiri dari encoder dan decoder. Encoder bertugas untuk memproses teks input dan menghasilkan representasi kontekstualnya. Sehingga encoder bertujuan untuk mengekstraksi informs penting dari teks input. Decoder bertugas untuk menghasilkan teks output dari hasil representasi oleh encoder. Teks akan dihasilkan secara bertahap dan akan bergantung dengan Langkah sebelumnya. Proses ini dilakukan secara berulang hingga teks output selesai terbentuk [4].

2.3. Workflow

Pada model Podcast Summarization memiliki beberapa tahapan sebagai berikut. Pada model ini, kami mengambil Podcast yang berasal dari youtube, sehingga Langkah pertama adalah mendownload pytube library untuk mendownload video youtube. Kemudian memasukkan link youtube pada variable Bernama youtube_video, dan mendownload dengan format MP4. Audio yang berhasil didownload disimpan dalam path "/content/audio.mp4/" setelah itu melakukan konversi file dari MP4 ke WAV file. Pada bagian Automatic Speech Recognition, install torch dan huggingsound. Kemudian melakukan pengecekan apakah GPU terdapat cuda. Jika ya maka nantinya akan menggunakan cuda, jika pada GPU tidak terdapat CUDA maka akan menggunakan CPU. Setelah itu panggil pretrained model wav2vec2. Tahapan berikutnya yaitu Audio Slicing, melakukan slicing dengan durasi 30 detik dan menetapkan audio frame = 16000. Kemudian setelah melakukan slicing, simpan file dengan format wav dan menyimpan hasil slicing dengan format wav.

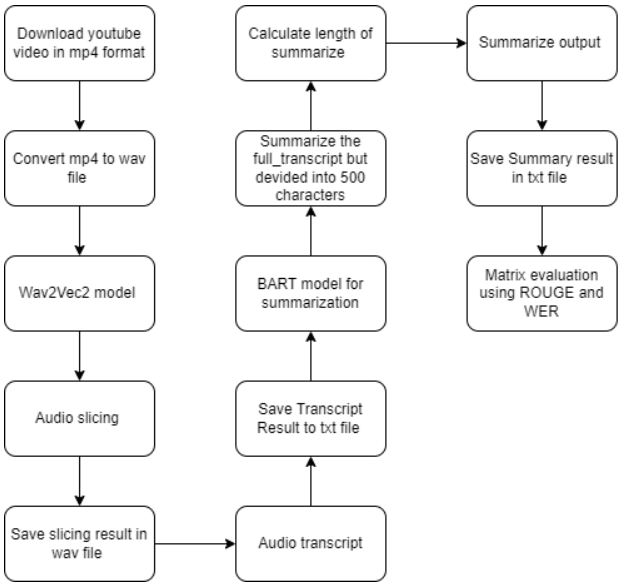

Figure 1. Podcast Summarization Workflow

Tahapan berikutnya setelah Audio Slicing adalah Audio Transcript, tahapan ini penting untuk dilakukan karena hasil transcript akan dilanjutkan pada bagian Text Summarizer untuk mendapatkan rangkuman dari podcast. Pada Audio Transcript menggunakan model wav2vec, kemudian menghitung Panjang character yang dihasilkan dari proses transcript, dan hasil dari Audio Transcript itu disimpan dengan format txt. Pada bagian text summarization, Langkah awal yang perlu dilakukan adalah install transformer dan import pipeline karena akan menggunakan function dari library pipeline. Model yang digunakan adalah BART, kemudian melakukan summarization. Setelah melakukan summarization hitung Panjang kata dari hasil summarization, tampilkan output dan simpan kedalam txt file. Langkah terakhir adalah menghitung Matrix Evaluation menggunakan WER dan ROUGE.

2.4. Evaluation Matrix Plan

Dalam Model Podcast Summarization yang kami buat, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) dan WER (Word Error rate) digunakan sebagai Matrix Evaluation untuk mengukur hasil ringkasan teks yang dihasilkan oleh model. ROUGE menghitung presisi, recall dan F1 score untuk mnegetahui unit teks yang tumpeng tindih. ROUGE digunakan untuk mengevaluasi kesamaan antara ringkasan teks yang dihasilkan oleh model dengan ringkasan yang dihasilkan oleh manusia. Semakin tinggi skor ROUGE maka akan semakin baik kualitas ringkasan yang dihasilkan [5].

Pada Automatic Speech Recognition Matrix Evaluation yang digunakan adalah WER (Word Error Rate). WER digunakan untuk mnegukur tingkat kesalahan dalam mengenali kata-kata dalam transkripsi audio. Dilakukan dengan mneghitung perbandingan antara jumlah kata salah yang dihasilkan oleh model dengan jumlah kata yang sebenarnya dari teks referensi. Semakin rendah score WER maka akan semakin baik dan akurat model dalam melakukan transkripsi audio [6].

3. HASIL DAN DISKUSI

3.1. Automatic Speech Recognition

Automatic Speech Recognition (ASR) merupakan proses mengubah speech signal menjadi trankripsi teks. ASR bertujuan untuk melakukan pemetaan Audio Signal, mengubah ucapan kedalam transkripsi teks yang berisi urutan kata [7]. Pada model yang kami buat, menggunakan pretrained model Wav2Vec2 dari Hugging sound. Sehingga sebelum melakukan inisialisasi pretrained Wav2Vec2, perlu dilakukan install torch dan huggingsound. Torch digunakan untuk pengecekan CUDA untuk mneggunakan GPU, karena diperlukan perangkat untuk menjalankan model. Jika tidak terdapat CUDA maka model akan dijalankan menggunakan CPU. Setelah pretrained model berhasil diinisialisasi, maka akan dilanjutkan dengan Audio Transcript. Pada tahapan ini, model akan dipanggil pada variable transcriptions. Kemudian hasil dari transkripsi akan dimasukkan pada variable baru yang bernama full_transcript kemudian disimpan dalam txt file yang nantinya akan digunakan untuk menghitung Evaluation Matrix menggunakan WER (Word Error Rate).

3.2. Text Summarization

Pada tahapan Text Summarization hal yang perlu dilakukan adalah install transformers karena kita akan

menggunakan library dari transformers. Kemudian inisialisasi pretrained model BART. Kemudian membuat variable baru Bernama summarized_text yang berguna sebagai tempat penyimpanan dari hasil summarization, lalu dilakukan summarization dengan membagi 500 karakter. Hasil dari ringkasan tersebut kemudian dihitung jumlah katanya lalu disimpan dalam bentuk txt file. Txt file ini berguna untuk perhitungan ROUGE pada bagian Evaluation Matrix.

3.3. Evaluation Matrix

Terdapat dua evaluation matrix yaitu ROUGE untuk mengevaluasi hasil dari Text Summarization menggunakan BART, hal ini dilakukan dengan menghitung tiga metrix evaluasi yaitu ROUGE 1, ROUGE 2 DAN ROUGE L.

	Rouge 1	**Rouge 2**	**Rouge L**
r	0.4768	0.3210	0.4495
p	0.5555	0.3315	0.5238
F1-score	0.5131	0.3262	0.4838

ROUGE 1 mengukur kesamaan kata per kata antara teks referensi dan teks hipotesis yang dihasilkan oleh model. Kemudian ROUGE 2 mengukur kesamaan pasangan kata antara teks referensi dengan teks hipotesis dan yang terakhir ROUGE L, akan mengukur kesamaan panjang substring antara teks referensi dan teks hipotesis. Pada teks referensi kami menggunakan hasil summarization yang dilakukan oleh manusia. Nilai recall (r) menunjukkan sejauh mana kata yang relevan dari teks referensi ditemukan dalam teks hipotesis. Kemudian nilai precision (p) menunjukkan sejauh mana kata yang ditemukan dalam teks hipotesis relevan dengan teks referensi. Dan F1-scores merupakan rata-rata dari recall dan precision yang memberikan perbandingan secara keseluruhan antara teks referensi dan teks hipotesis. Pengukuran ini dilakukan untuk menilai sejauh mana teks hipotesis yang dihasilkan oleh model cocok dengan teks referensi yang sebenarnya. Semakin tinggi nilai ROUGE maka kualitas summarization yang dihasilkan oleh model akan semakin baik.

Hasil pada tabel menunjukkan bagian ROUGE 1 nilai recall sebesar 47.68% kata yang relevan dari teks referensi berhasil ditemukan dalam hasil ringkasan yang dihasilkan oleh model. Nilai precision sebesar 55.66% dalam hasil ringkasan yang dihasilkan oleh model adalah relevan dengan teks referensi. F1 score sebesar 51.32% merupakan rata-rata harmonic dari recall dan precision, F1-score memberikan perbandingan keseluruhan antara teks referensi dan hasil summarization yang dihasilkan oleh model. Lalu pada ROUGE 2 yang mengukur kesamaan pasangan kata, pada recall mendapat score sebesar 32.10% dimana bigram (pasangan kata) yang relevan dari teks referensi berhasil ditemukan pada hasil ringkasan. Precision sebesar 33.16% pasangan kata pada hasil summarization relevan dengan teks referensi. F1-score sebesar 32.62% merupakan rerata harmonic dari recall dan precision. ROUGE L mengukur kesamaan substring terpanjang yang berurutan. Nilai recall sebesar 44.96% substring terpanjang yang relevan dari teks referensi berhasil ditemukan pada hasil ringkasan oleh model. Precision sebesar 52.38% substring terpanjang pada hasil ringkasan relevan dengan teks referensi. F1-score sebesar 48.39% merupakan rata-rata nilai recall dan precision.

Word Error Rate (WER) merupakan matrik evalusi yang digunakan untuk mengukur tingkat kesalahan dalam Automatic Speech Recognition). WER didapatkan dengan menghitung eprbandingan antara jumlah kata yang salah yang dihasilkan oleh model, dibandingkan dengan hasil transkripsi sebenarnya, kami menggunakan hasil transkripsi dari API youtube sebagai teks referensi. Pada model ini dihasilkan nilai WER sebesar 26.67% dimana setiap 100 kata yang sebenarnya dalam transkripsi, system ASR menghasilkan sekitar 26 kata yang salah. Semakin rendah nilai WER maka semakin baik kinerja ASR dimana mampu menghasilkan transkripsi yang lebih akurat dan mendekati teks sebenarnya. Sehingga nilai WER ini dapat digunakan sebagai indicator mengenai kualitas dari system ASR dalam mengenail dan mentranskripsi sinyal suara.

4. KESIMPULAN

Berdasarkan hasil uji coba mengubah audio podcast dari youtube menjadi teks transkripsi menggunakan model Wav2Vec2 dapat diambil kesimpulan bahwa model dapat menghasilkan teks trankripsi dengan Word Error Rate (WER) sebesar 26,67%. Di sisi lain, uji coba meringkas teks transkripsi podcast menggunakan model BART menghasilkan skor ROUGE 1 sebesar 51,32%, ROUGE 2 sebesar 32,62%, dan ROUGE L sebesar 48,39%. Dari hasil WER dan ROUGE yang telah didapatkan, bisa disimpulkan bahwa model ini mampu menghasilkan teks transkripsi yang cukup akurat dan teks ringkasan yang relevan terhadap teks aslinya. Meski menunjukkan hasil yang cukup baik, kualitas transkripsi dan ringkasan diharapkan dapat diperbaiki lagi dengan meningkatkan kinerja dari masing-masing model.

DAFTAR PUSTAKA

[1] S. Kaiqiang , L. Chen, W. Xiaoyang, Y. Dong and L. Fei , "Towards Abstractive Grounded Summarization of Podcast Transcripts," *Association for Computational Linguistics*, vol. 1, p. 4407, 2022.

[2] Hugging Face, "Datasets : common_voice," [Online]. Available: https://huggingface.co/datasets/common_voice. [Accessed 2 6 2023].

[3] Qui Shi ZHu, Jie Zhang, ZI Qiang Zhang, Ming Hui Wu, Xin Fang and Li Rong Dai, "A Noise Robust Self Supervised Pre Training Model Based Speech Representation Learning For Automatic Speech Recognition," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 2, 2022.

[4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and Z. Luke, "BART : Denoising Sequence-to-Sequence Pre-training for Natural Language Generator, Translation and Comprehension," p. 7872, 2020.

[5] Y. Ruifeng, W. Zili and L. Wenjie , "Fact-level Extractive Summarization with Hierarchical Graph Mask on BERT," *Xidian University*, p. 5636, 202.

[6] S. Roy, "Semantic-WER : A Unified Metric For The Evaluation Of ASR Transcript For End Usability," p. 2, 2020.

[7] S. Karpagavalli and E. Chandra , "A Review on Automatic Speech Recognition Architecture and Approaches," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, p. 2, 2016.