

LAPORAN TUGAS BESAR
NATURAL LANGUAGE PROCESSING HUGGING FACE CHAPTER
1 - 4

MATA KULIAH MACHINE LEARNING



Disusun oleh:
Alfikri (1103223015)

PROGRAM STUDI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
TELKOM UNIVERSITY BANDUNG

Chapter 1 NLP Hugging Face

Pustaka Transformers yang dikembangkan oleh Hugging Face menyediakan berbagai alat untuk pemrosesan bahasa alami (Natural Language Processing/NLP) menggunakan model pembelajaran mesin yang canggih. Salah satu fitur utama dalam pustaka ini adalah fungsi `pipeline()`. Fungsi ini menghubungkan model dengan langkah-langkah preprocessing dan postprocessing yang diperlukan, memungkinkan pengguna untuk langsung memasukkan teks dan mendapatkan keluaran berupa hasil analisis atau prediksi yang mudah dipahami.

Fungsi `pipeline()` dapat digunakan dengan sangat sederhana. Langkah pertama adalah mengimpor pustaka dan menginisialisasi pipeline. Berikut adalah contoh penggunaannya:





```
[ ] from transformers import pipeline
```

Untuk analisis sentimen, pipeline dapat diinisialisasi sebagai berikut:

```
[ ] classifier = pipeline("sentiment-analysis")
classifier("I've been waiting for a HuggingFace course my whole life.")
```

Hasilnya akan menunjukkan label sentimen dari teks tersebut, seperti "POSITIVE" dengan tingkat akurasi tersebut. Pipeline ini juga mendukung analisis beberapa teks sekaligus:

```
classifier = pipeline("sentiment-analysis")
classifier("I've been waiting for a HuggingFace course my whole life.")
```

No model was supplied, defaulted to `distilbert/distilbert-base-uncased-finetuned-sst-2-english` and revision `714eb0f` (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).
Using a pipeline without specifying a model name and revision in production is not recommended.
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret 'HF_TOKEN' does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
config.json: 100%  629/629 [00:00<00:00, 36.9kB/s]
model.safetensors: 100%  268M/268M [00:02<00:00, 168MB/s]
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 2.08kB/s]
vocab.bpe: 100%  232k/232k [00:00<00:00, 2.73MB/s]
Device set to use cpu
[{'label': 'POSITIVE', 'score': 0.9998049521446228}]

Salah satu keunggulan utama pustaka ini adalah kemampuannya untuk melakukan klasifikasi zero-shot. Dengan pipeline ini, pengguna dapat menentukan label klasifikasi secara langsung tanpa bergantung pada label yang digunakan saat model dilatih. Hal ini memungkinkan fleksibilitas dalam berbagai kasus penggunaan.

```
[ ] from transformers import pipeline

[ ] classifier = pipeline("zero-shot-classification")
classifier(
    "This is a course about the Transformers library",
    candidate_labels=["education", "politics", "business"],
)

No model was supplied, defaulted to facebook/bart-large-mnli and revision d7645e1 (https://huggingface.co/facebook/bart-large-mnli).  
Using a pipeline without specifying a model name and revision in production is not recommended.  
Device set to use cpu  
{'sequence': 'This is a course about the Transformers library',  
 'labels': ['education', 'business', 'politics'],  
 'scores': [0.8445994257926941, 0.11197380721569061, 0.04342673346400261]}
```

Pustaka Transformers juga mendukung Text Generation. Dalam penggunaan ini, pipeline menerima sebuah prompt dan secara otomatis melengkapinya dengan teks yang sesuai.

```
[ ] from transformers import pipeline
```

```
[ ] generator = pipeline("text-generation")  
generator("In this course, we will teach you how to")
```

⚠ No model was supplied, defaulted to `openai-community/gpt2` and revision `007a3b8` (<https://huggingface.co/openai-community/gpt2>).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use `cpu`.
Setting `'pad_token_id'` to `'eos_token_id':50256` for open-end generation.
[{"generated_text": "In this course, we will teach you how to implement data in the real world to create your own data models. In particular, we will help you design better, more powerful, more readable data objects. You will be taught the same concepts used by"}]

Selain analisis sentimen, zero-shot classification, dan text generation, pustaka Hugging Face Transformers menawarkan berbagai fitur lainnya yang mendukung tugas-tugas kompleks dalam pemrosesan bahasa alami. Fitur-fitur tersebut mencakup mask filling, summarization, named entity recognition (NER), question answering, translation, serta pembahasan mengenai bias dan keterbatasan model.

Mask filling adalah fitur yang memungkinkan model memprediksi kata atau token yang hilang dalam sebuah teks. Fitur ini sangat bermanfaat untuk menyempurnakan kalimat dengan struktur yang tidak lengkap, seperti mengisi kekosongan dalam dokumen atau menyarankan kata-kata yang relevan dalam sebuah konteks. Model yang digunakan dalam pipeline ini dilatih untuk mengidentifikasi kata-kata yang paling mungkin mengisi posisi yang kosong, berdasarkan pola-pola dalam data pelatihan.

Fitur summarization digunakan untuk merangkum teks yang panjang menjadi versi ringkas yang tetap mencakup informasi utama. Dengan menggunakan model yang dirancang untuk memahami konteks dokumen secara keseluruhan, pipeline ini memproses teks input dan menghasilkan keluaran berupa ringkasan yang singkat namun informatif. Teknologi ini sering diterapkan dalam pembuatan ringkasan artikel, laporan, atau dokumen teknis.

Named Entity Recognition (NER) adalah fitur yang memungkinkan identifikasi entitas tertentu dalam sebuah teks, seperti nama orang, lokasi, organisasi, atau waktu. Fitur ini dapat digunakan untuk mengekstrak informasi penting dari dokumen, sehingga sangat berguna dalam aplikasi seperti analisis data berita, sistem pencarian dokumen, atau pemrosesan formulir otomatis.

Fitur question answering memberikan kemampuan kepada model untuk menjawab pertanyaan berdasarkan teks yang diberikan sebagai konteks. Dalam implementasinya, model membaca teks yang relevan, memahami isinya, dan menghasilkan jawaban yang paling sesuai. Teknologi ini sering digunakan dalam aplikasi chatbot, pencarian berbasis teks, atau sistem pendukung keputusan berbasis dokumen.

Fitur penerjemahan memungkinkan transformasi teks dari satu bahasa ke bahasa lain dengan akurasi tinggi. Model dalam pipeline ini dilatih menggunakan data multibahasa, sehingga dapat menghasilkan terjemahan yang mendekati makna aslinya. Fitur ini mendukung berbagai pasangan bahasa dan dapat diterapkan dalam aplikasi lintas budaya atau layanan global.

Penting untuk memahami bahwa model dalam pustaka Transformers dilatih menggunakan data yang berasal dari dunia nyata, yang mungkin mengandung bias bawaan. Sebagai contoh, bias gender atau etnis dapat muncul dalam hasil keluaran model jika data pelatihan mengandung pola-pola yang merepresentasikan stereotip tertentu. Oleh karena itu, pengguna disarankan untuk memonitor hasil yang dihasilkan oleh model, memahami konteks penggunaan, dan mempertimbangkan langkah-langkah mitigasi bias. Ini termasuk melakukan evaluasi kritis terhadap model sebelum diimplementasikan dalam skenario produksi, terutama jika model akan memengaruhi keputusan yang sensitif.

Chapter 2 NLP Hugging Face

Pendekatan modern dalam Natural Language Processing (NLP) telah berkembang pesat dengan adanya pipeline yang mengintegrasikan proses pengolahan data, pemrosesan model, dan interpretasi hasil. Salah satu platform utama yang mendukung pendekatan ini adalah pustaka 'transformers' dari Hugging Face. Pipeline ini dirancang untuk menyederhanakan alur kerja, sehingga mempermudah pengguna dalam membangun dan menerapkan model berbasis deep learning.

Tahap awal dalam pipeline NLP melibatkan preprocessing, di mana data mentah, seperti teks, dikonversi menjadi format numerik yang dapat diterima oleh model. Proses ini sering mencakup langkah-langkah seperti tokenisasi dan normalisasi teks, yang memastikan data sesuai dengan format model.

Selanjutnya, data yang telah diproses dimasukkan ke dalam model pada tahap model input. Pada tahap ini, model menjalankan algoritma untuk menganalisis data dan menghasilkan keluaran, seperti prediksi atau representasi vektor.

Tahap akhir, yaitu postprocessing, bertujuan untuk mengubah keluaran mentah model menjadi informasi yang bermakna. Contohnya termasuk mengonversi vektor keluaran menjadi kalimat yang dapat dibaca atau mengelompokkan teks berdasarkan emosi tertentu.

Dengan struktur tiga langkah ini, pipeline NLP menjadi alat yang sangat efisien dalam menangani berbagai aplikasi, mulai dari analisis sentimen hingga penerjemahan teks. Untuk mendukung pemahaman ini, visualisasi alur kerja sering digunakan untuk memperlihatkan hubungan antara tahap preprocessing, pemrosesan model, dan postprocessing, yang menunjukkan integrasi proses secara keseluruhan.

Chapter 3 NLP Hugging Face

Proses dimulai dengan pelatihan sederhana menggunakan PyTorch, di mana pasangan kalimat tokenisasi dan label diterapkan untuk menghitung nilai kerugian (loss) melalui backward propagation. Pendekatan ini diilustrasikan sebagai langkah awal dalam memahami arsitektur model sequence classifier.

Setelah itu, pengenalan terhadap dataset MRPC mengungkapkan bahwa dataset ini terdiri dari pasangan kalimat dengan label yang menentukan apakah kedua kalimat tersebut adalah parafrasa. Dataset ini dipilih karena ukurannya yang kecil, memungkinkan eksperimen cepat dalam proses pelatihan.

Langkah selanjutnya adalah preprocessing dataset dengan mengubah teks menjadi representasi numerik menggunakan tokenizer. Model BERT, misalnya, mengharuskan input berupa pasangan kalimat dengan token khusus seperti [CLS] untuk menunjukkan awal input dan [SEP] untuk memisahkan kalimat. Teknik seperti padding dinamis diperkenalkan untuk mengoptimalkan pemrosesan batch dalam pelatihan.

Fine-tuning model dilakukan dengan menggunakan Trainer API yang disediakan oleh Hugging Face, yang menyederhanakan proses pelatihan dengan menangani hyperparameter, data loader, dan pipeline evaluasi. Proses ini termasuk pengaturan metrik evaluasi seperti akurasi dan F1-score, yang relevan untuk menilai performa model pada dataset MRPC.

Selain API Trainer, loop pelatihan manual juga dijelaskan untuk memberikan fleksibilitas lebih. Pengaturan ini mencakup definisi optimizer, scheduler pembelajaran, dan integrasi GPU untuk mempercepat pelatihan.

Pada tahap evaluasi, prediksi dari model dibandingkan dengan label menggunakan pustaka Evaluate dari Hugging Face. Proses ini memungkinkan analisis lebih rinci terhadap kinerja model, seperti akurasi dan F1-score, untuk setiap epoch pelatihan.

Chapter 4 NLP Hugging Face

Hugging Face menyediakan berbagai model NLP pretrained yang dapat langsung digunakan untuk berbagai tugas, seperti pengisian kata kosong (*mask filling*). Salah satu model yang digunakan adalah CamemBERT, yang dirancang untuk bahasa Prancis. Model ini dapat dengan mudah diakses melalui fungsi pipeline, yang menyederhanakan proses penggunaan model.

Auto Classes, seperti AutoTokenizer dan AutoModel, memberikan fleksibilitas dalam bekerja dengan berbagai checkpoint model tanpa tergantung pada arsitektur spesifik. Hal ini mempermudah pergantian antar model dalam aplikasi NLP.

- Implementasi dengan Auto Classes:

```
from transformers import AutoTokenizer, AutoModelForMaskedLM
tokenizer = AutoTokenizer.from_pretrained("camembert-base")
model = AutoModelForMaskedLM.from_pretrained("camembert-base")
```

Model yang telah dilatih atau dimodifikasi dapat diunggah ke Hugging Face Hub menggunakan API `push_to_hub`. Ini memungkinkan model untuk dibagikan dengan komunitas atau digunakan kembali dalam proyek lain.

- Contoh implementasi:

```
model.push_to_hub("nama-model")
tokenizer.push_to_hub("nama-model")
```

Rintangan yang dihadapi

Pada chapter 3 terdapat config pada accelerate, ketika melakukan proses config terjadi beberapa error yang disebabkan ketidaksesuaian code dengan lingkungan konfigurasi, sehingga solusinya ialah menyesuaikan code dengan lingkungan accelerate yang dipilih

```
[ ] accelerate config

-----In which compute environment are you running?
Please input a choice index (starting from 0), and press enter
→ This machine
   AWS (Amazon SageMaker)

This machine
-----Which type of machine are you using?
Please input a choice index (starting from 0), and press enter
→ No distributed training
   multi-CPU
   multi-XPU
   multi-GPU
   multi-NPU
   multi-MLU
   multi-MUSA
   TPU
TPU
No distributed training
Do you want to run your training on CPU only (even if a GPU / Apple Silicon / Ascend NPU device is available)? [yes/NO]:yes
Do you want to use Intel PyTorch Extension (IPEX) to speed up training on CPU? [yes/NO]:yes
Do you wish to optimize your script with torch dynamo?[yes/NO]:yes
-----Which dynamo backend would you like to use?
Please input a choice index (starting from 0), and press enter
eager
aot_eager
→ inductor
aot_ts_nvfuser
nvprims_nvfuser
cudagraphs
ofi
fx2trt
onnxrt
tensorrt
aot_torchxla_trace_once
torchxla_trace_once
ipex
tvm
```

Lalu, pada chapter 4 diharuskan melakukan push ke hub API Hugging Face, Pada saat itu terjadi beberapa kali error ketika melakukan pembuatan akun Hugging Face, yang ternyata diharuskan menggunakan akun mahasiswa untuk mendaftarkannya. Sehingga ketika proses itu berhasil dilakukan, maka push ke hub dapat berjalan dengan baik.

```
[ ] model.push_to_hub("dummy-model")

model.safetensors: 100% ██████████ 443M/443M [00:08<00:00, 51.7MB/s]
CommitInfo(commit_url='https://huggingface.co/Lakuna/dummy-model/commit/d196039128ff52dbfe8c38ee73e
repo_url=RepoUrl('https://huggingface.co/Lakuna/dummy-model', endpoint='https://huggingface.co', re

[ ] tokenizer.push_to_hub("dummy-model")

README.md: 100% ██████████ 5.17k/5.17k [00:00<00:00, 74.5kB/s]
sentencepiece.bpe.model: 100% ██████████ 811k/811k [00:00<00:00, 3.19MB/s]
CommitInfo(commit_url='https://huggingface.co/Lakuna/dummy-model/commit/6c005efd2d58a7983bf10b7e80e
repo_url=RepoUrl('https://huggingface.co/Lakuna/dummy-model', endpoint='https://huggingface.co', re
```