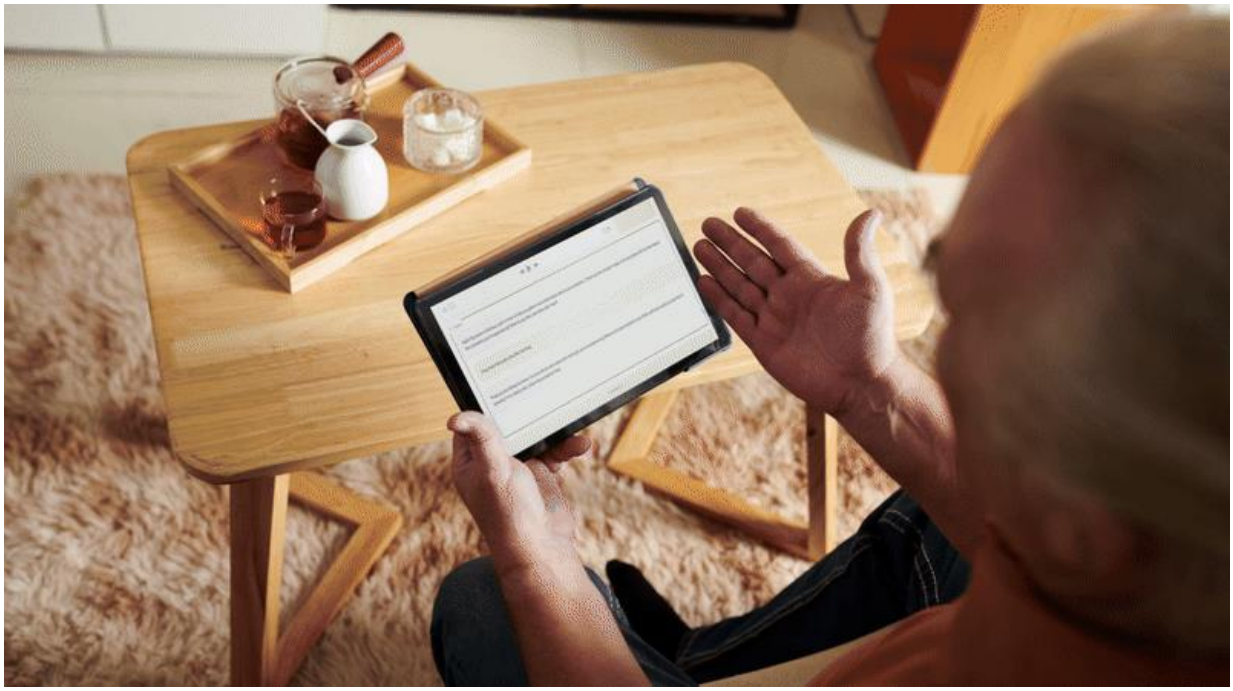
パート 1: OpenVINO™ を使用した AI 医療アシスタントの作成: ヘルスケアの変革

この記事は、Medium に公開されている「Part One: Crafting an AI-Powered Medical Assistant: Transforming Healthcare with OpenVINO™」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

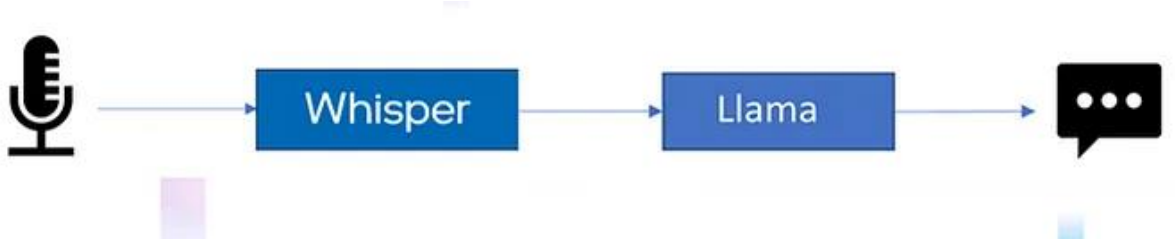
誰もが、病院の待合室で座っているとき、時間が経つごとに不安が増していくのを経験したことがあるでしょう。特に深刻な病気や慢性疾患を抱える患者にとっては、この待ち時間は非常にストレスになります。一方、医師も、過密なスケジュールの中、限られた時間で適切な治療を行わなければならないという課題に直面しています。AI を活用して、この状況を改善する方法を考えてみましょう。患者が診察室に入る前に対応してくれる AI アシスタントを思い浮かべてみてください。このアシスタントは患者の懸念事項に耳を傾け、症状や病歴に関する包括的な情報を収集して、医師に詳細な要点を提供します。患者が医師の診察を受けるまでに準備が整っているため、焦点を絞った診察を迅速に行うことができます。



このような AI アシスタントを作成するには、自動音声認識 (ASR) や大規模言語モデル (LLM) など、複数の高度な AI モデルをデプロイする必要があります。これらのモデルは計算集約型であり、効率良く実行するには多くのリソースが必要です。最適化しないでこれらのモデルをリアルタイムで実行すると、時間がかかり、リソースを大量に消費する可能性が高くなるため、ペースの速い医療現場では使い物にならないでしょう。ここで、モデルの最適化と効率的なデプロイが重要になります。OpenVINO™ ツールキットを活用すれば、これらの課題に対処できます。この強力なフレームワークは、インテルのハードウェア上で AI モデルを最適化してデプロイし、パフォーマンスを向上させて、リソースを効率的に利用できるようにします。また、これらのモデルをさまざまなデバイスにデプロイする柔軟性を提供し、実際のアプリケーションで利用できる汎用的なソリューションにします。では、OpenVINO™ ツールキットを使用してカスタム AI 医療アシスタントを作成する方法を紹介しましょう。

実装の手順

前述したように、この AI アシスタントの機能を考えると、ワークフローは次のようになります。



環境を設定してアプリケーションを実行する手順を次に示します。

1. ステージの設定

まず最初に、環境を設定します。アシスタントは Python* 3.8 以降で動作します。以下の説明では、Ubuntu* と Windows* の設定について取り上げます。

Ubuntu* の場合: 必須ライブラリーとツールをインストールします。

```
sudo apt install git gcc python3-venv python3-dev
```

注: Windows* を使用している場合は、Microsoft* Visual C++* 再頒布可能パッケージのダウンロード・ページから、X64 再頒布可能パッケージ (vc_redist.x64.exe) をダウンロードしてインストールしてください。

2. 仮想環境の作成

クリーンで管理しやすい状態を保つため、仮想環境を作成します。この分離された環境により、依存関係が適切に保持されます。

Ubuntu* の場合:

```
python3 -m venv venv
source venv/bin/activate
```

Windows* の場合:

```
python3 -m venv venv
venv\Scripts\activate
```

3. リポジトリーのクローンの作成

次に、プロジェクトが格納されるリポジトリーのクローンを作成します。この手順は、どちらのオペレーティング・システムでも同じです。

```
git clone https://github.com/openvinotoolkit/openvino_build_deploy.git
cd openvino_build_deploy/ai_ref_kits/custom_ai_assistant
```

4. 依存関係のインストール

仮想環境を有効にしたら、必要なパッケージをインストールします。

```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

5. モデルのアクセスと設定

自然言語の理解には、Meta の Llama モデルを利用します。モデルにアクセスするには、Hugging Face による認証が必要です。

```
huggingface-cli login
```

プロンプトに従って、Meta AI の Web サイトで使用したものと同じメールアドレスを使用して認証します。Llama モデルをダウンロードして使用するには、この手順が重要です。

6. OpenVINO™ を使用した変換と最適化

モデルを実際のアプリケーションで効率的に使用するには、モデルを変換して最適化する必要があります。

- **自動音声認識 (ASR) モデル:**

```
python convert_and_optimize_asr.py --asr_model_type distil-whisper-large-v2 --precision int8
```

このスクリプトは、ASR モデルを変換して最適化し、重みの量子化を行ってパフォーマンスを向上させます。

- **チャットモデル (Llama):**

```
python convert_and_optimize_chat.py --chat_model_type llama3-8B --precision int4
```

このスクリプトは、重みの量子化を行って、チャットモデルがインテルのハードウェア上で効率良く動作するようにします。

7. アプリケーションの実行

モデルの準備ができたら、AI アシスタントと対話するためのユーザーフレンドリーなインターフェイス、Gradio を使用してアプリケーションを起動します。

```
python app.py --asr_model_dir path/to/asr_model --chat_model_dir path/to/chat_model
```

アシスタントとの対話には、ローカル URL (通常は `http://127.0.0.1:XXXX`) を使用します。パブリックアクセスの場合は、`--public_interface` オプションを指定します。

Gradio URL に移動すると、マイクのアイコンが付いたインターフェイスが表示されます。アイコンをクリックして質問を行い、アシスタントがテキストを処理して応答するのを確認します。このインタラクティブなエクスペリエンスは、アシスタントが意味のある対話を理解して処理する能力を示すものです。

8. 多言語サポートの拡張

OpenVINO™ は生成 AI モデルの推論を最適化および高速化する広範なサポートを提供しているため、英語だけでなくほかの言語をサポートするように AI アシスタントを簡単に拡張できます。例えば、次の手順は中国語で動作するように AI アシスタントを拡張する方法を示しています。

ASR およびチャットの新しいモデルの追加: ASR モデルを変更するには、次の操作を行います。

- **`MODEL_MAPPING` の変更:** 目的のモデルを `MODEL_MAPPING` 辞書に追加します。

```
MODEL_MAPPING = {
    "distil-whisper-large-v2": "distil-whisper/distil-large-v2",
    "new-model": "path/to/your/new-model",
}
```

- モデル選択の構成を**変更**します。

```
parser.add_argument("--asr_model_type", type=str, choices=["distil-whisper-
large-v2", "path/to/your/new-model"],
                        default="distil-whisper-large-v2", help="Speech
recognition model to be converted")
```

- 音声認識の ASR モデルを中国語で実行するには、次のコマンドを使用します。

```
python convert_and_optimize_asr.py --asr_model_type belle-distilwhisper-
large-v2-zh --precision int8
```

- 同様に、チャットモデルに新しいモデルを追加し、チャットモデルを中国語で実行するには、次のコマンドを使用します。

```
python convert_and_optimize_chat.py --chat_model_type qwen2-7B --precision
int4
```

- 最後に、AI アシスタントを中国語で動作させるには、次のコマンドを使用して `app.py` を実行します。

```
python app.py --asr_model_dir path/to/belle-distilwhisper-large-v2-zh --
chat_model_dir path/to/qwen2-7B
```

まとめ

AI 医療アシスタントを起動して実行するための基本的な手順はこれで完了です。このアシスタントを異なる業界向けにカスタマイズし、基本キットを使用してさまざまなモデルを統合する方法を学習したい場合は、パート 2 の詳細なヒントと追加のガイダンスを参照してください。コーディングを楽しみましょう!

関連情報

パート 2: OpenVINO™ を使用した AI 医療アシスタントのカスタマイズ
エッジ AI リファレンス・キット (英語)
OpenVINO™ モデルサーバー GitHub* リポジトリー (英語)
OpenVINO™ ドキュメント (英語)
Jupyter* Notebook (英語)
インストールとセットアップ (英語)
製品ページ (英語)

著者紹介

Anisha Udayakumar (英語) は、インテル コーポレーションの AI ソフトウェア・エバンジェリストで、OpenVINO™ ツールキットを担当しています。インテルでは、OpenVINO™ の機能を紹介することにより、開発者コミュニティーを強化し、開発者が AI プロジェクトを改善できるように支援しています。インドの大手 IT 企業でイノベーション・コンサルタントを務めた経歴を持ち、革新的なソリューションのために新しいテクノロジーを活用することをビジネスリーダーに伝えてきました。その専門知識は AI、クロス・リアリティー、5G にまで及び、特にコンピューター・ビジョンに情熱を注いでいます。世界的な小売クライアント向けに、サステナビリティーの目標を推進するビジョンベースのアルゴリズム・ソリューションを開発したこともあります。彼女は生涯学習者かつイノベーターであり、テクノロジーの変革の影響を探求して共有することに専念しています。

Zhuo Wu は、インテル コーポレーションの AI エバンジェリストで、OpenVINO™ ツールキットを担当しています。研究対象はディープラーニング・テクノロジーから 5G 無線通信技術まで多岐にわたり、コンピューター・ビジョン、マシンラーニング、エッジ・コンピューティング、IoT システム、無線通信物理層アルゴリズムに貢献してきました。これまで、自動車、銀行、保険などさまざまな業界の企業に、エンドツーエンドのマシンラーニングおよびディープラーニング・ベースのソリューションを提供してきました。4G-LTE および 5G 無線通信システムに関する広範な研究も行っており、中国のベル研究所で研究員として勤務していたときに複数の特許を取得しています。上海大学の准教授時代には、主任研究者としていくつかの研究プロジェクトを先導しました。

OpenVINO™ ツールキットとは

AI を加速する無償のツールである OpenVINO™ ツールキットは、インテルが無償で提供しているインテル製の CPU や GPU、VPU、FPGA などのパフォーマンスを最大限に活用して、コンピューター・ビジョン、画像関係をはじめ、自然言語処理や音声処理など、幅広いディープラーニング・モデルで推論を最適化し高速化する推論エンジン/ツールスイートです。

OpenVINO™ ツールキット・ページでは、ツールの概要、利用方法、導入事例、トレーニング、ツール・ダウンロードまでさまざまな情報を提供しています。ぜひ特設サイトにアクセスしてみてください。

https://www.intel.co.jp/content/www/jp/ja/internet-of-things/openvino-toolkit.html

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティーを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。