

Guide d'utilisation de l'implémentation du serveur NISQA pour Debian

Auteurs :

Séverin CHEVALIER

Liam COURSDON

Jules DECAESTECKER

Zaccarie KANIT

Clémence MARINIER

Léo STENGEL

Elias TRANCHANT

Décembre 2022

1 Présentation

Ce projet a été réalisé par 7 étudiants d'IMT Atlantique Brest avec Vincent BARRIAC, Yvain JORIGNY et Nicolas PENNANEACH de Orange (Innovations/Networks). NiSQA est un algorithme évaluant la qualité d'une communication audio développé par des chercheurs de l'Université Technique de Berlin. Ce projet visait à remplacer l'algorithme PESQ anciennement utilisé. Ce guide d'utilisation détaille l'architecture et le fonctionnement du logiciel Debian et complète le readme associé à ce projet.

2 Contenu de l'archive

Dans cette archive se trouve plusieurs fichier :

- README.md : description du répertoire et consignes d'utilisation
- nisqacli : dossier contenant le package debian. A packager pour créer un fichier .deb qui sera ensuite installé
- requirements.txt : fichier contenant les dépendances python à installer
- install.sh : script bash installant les dépendances et le logiciel debian
- dg105.wav : fichier audio servant aux tests
- test.sh : script bash effectuant un test du bon fonctionnement du logiciel debian en utilisant l'audio dg105.wav

3 Installation

3.1 Automatisée

En se plaçant dans le répertoire du projet, l'exécution, avec les droits root, du script install.sh se chargera de télécharger les dépendances et d'installer le logiciel :

```
sudo ./install.sh
```

3.2 Manuelle

En se plaçant dans le répertoire du projet :

- Pour construire le .deb:

```
dpkg-deb -b nisqa
```

- Pour installer le logiciel sans installer les dépendances:

```
sudo dpkg -i nisqa.deb
```

- Pour télécharger et installer les librairies et le logiciel :

```
sudo pip install -r requirements.txt
```

```
sudo apt-get install ./nisqa.deb
```

4 Utilisation

4.1 Lancement d'une mesure

- Pour utiliser avec un fichier .wav

```
su - nisqa -c "nisqacli" < audio.wav
```

- Test avec test.sh

```
./test.sh
```

Ce script fait le test avec le fichier dg105.wav et compare le résultat obtenu avec le résultat qu'on obtient normalement si on applique directement le modèle NiSQA au fichier audio (écrit à la main dans le script). Comme l'algorithme NiSQA est du machine learning, les décimales des résultats peuvent différer selon les machines. C'est pourquoi on ne compare que les 3 premiers chiffres après la virgule.

4.2 Résultats attendus

Le résultat des commandes précédentes sont de la forme :

```
[MOS, Noisiness, Coloration, Discontinuity, Loudness]
```

Chaque note est comprise entre 1 et 5. On peut avoir par exemple :

```
[1.495162, 2.407247, 2.001792, 2.026033, 2.895231]
```

5 Fonctionnement et Maintenance

5.1 Algorithme Nisqa

Le modèle d'évaluation de la qualité audio que nous utilisons est nommé NiSQA pour Non-intrusive Speech Quality Assessment. C'est un réseau de neurones convolutif basé donc sur des méthodes de deep-learning codé en python. Il a été entraîné sur différents ensembles de données et est utilisable tel quel.

Pour l'instant, l'algorithme fonctionne sans fichier de référence. Cela signifie que le fichier audio dégradé après transmission n'est pas comparé avec celui enregistré avant la transmission. Les sources sont hébergées sur le dépôt Github de G. Mittag : <https://github.com/gabrielmittag/NISQA>

5.2 Architecture du logiciel

Cette partie décrit les différents composants du logiciel Debian pour que par la suite, si une nouvelle version du logiciel NiSQA est publié ou si Orange décide de changer d'algorithme d'évaluation, la modification puisse se faire sans problème.

5.2.1 Le logiciel Debian et son installation

Le répertoire nisqa est composé de 2 sous-répertoires :

- DEBIAN : Ce répertoire contient les informations propres à l'installation du logiciel Debian réalisé lors de "dpkg -i nisqa"
- home/nisqa : Lors de l'installation du package Debian, un utilisateur nisqa est créé. Le contenu du répertoire home/nisqa sera copié dans /home/nisqa

Le dossier DEBIAN contient 4 fichiers :

- control : Description des dépendances du logiciel, de son nom, des auteurs, ...
- postint : Script lancé après le dépaquetage du logiciel. C'est lui qui crée l'utilisateur nisqa.
- preinst : Script lancé avant le dépaquetage.
- prerm : Si le logiciel est supprimé, ce script est exécuté juste avant.

5.2.2 Utilisation de la ligne de commande

Lors de la commande 'su - nisqa -c "nisqacli" < audio.wav', on se fait passer pour l'utilisateur nisqa (qui n'a pas de mot de passe) et on exécute la commande nisqacli... Cette commande appelle le script bash nisqacli contenu dans home/nisqa/bin/. C'est ce fichier qui va s'occuper d'appeler le script python de NiSQA et d'afficher la réponse.

5.2.3 L'évaluation de la qualité audio : NiSQA

Comme dit précédemment, l'appel du script python est effectué dans home/nisqa/bin/nisqacli :

```
x="$(python3 bin/nisqa/run_predict.py --mode predict_file --pretrained_model bin/nisqa/weights/nisqa_model/nisqa_model.tar.gz audio.wav)"
```

Dans le dossier home/nisqa/bin se trouve un dossier nisqa_model qui est le clone du répertoire du github de Gabriel Mittag. C'est ce dossier qu'il faut mettre à jour si l'algorithme vient à évoluer. Comme vu dans la commande ci-dessus, le script run_predict.py est appelé et utilise les poids contenu dans weights/nisqa.tar .

Les erreurs sont redirigées sur la sortie standard de sorte que l'utilisateur du logiciel puisse connaître l'origine de l'erreur.