CORRECTION DE LA PAROLE

Elise Beaussart, Hugo Breniaux

Université Laval
Département d'informatique et de génie logiciel, Faculté des Sciences et de Génie
Répertoire Github : https://github.com/Yugoow/MLSP_Project
{elise.beaussart.1, hugo.breniaux.1}@ulaval.ca

RÉSUMÉ

La clé de l'apprentissage efficace d'une langue étrangère réside dans la pratique de l'oral. Nous avons entrepris le développement d'un Proof of Concept (POC) en utilisant des modèles d'intelligence artificielle, capables de générer un audio corrigé à partir d'un enregistrement oral en anglais. À l'état actuel de la recherche, aucune référence à un outil similaire au nôtre n'a été identifiée. De ce fait, pour concrétiser ce projet nous avons effectué des évaluations individuelles, en nous appuyant sur des modèles existants et performants afin de perfectionner la mise en œuvre de notre pipeline. Whisper a été sélectionné comme modèle de reconnaissance automatique de la parole (ASR) et a été adapté pour répondre spécifiquement à notre cas d'utilisation. Nous supposons ici des utilisateurs ayant l'indien comme langue maternelle et présentant un certain accent lorsqu'ils parlent anglais. Quant à T5, il a été choisi et affiné de manière spécifique pour les tâches de correction des erreurs grammaticales (GEC). Enfin, le modèle de synthèse vocale gTTS a été retenu en raison de sa simplicité et de sa rapidité d'exécution.

Termes d'indexation— apprentissage, langue, oral, IA, correction grammaticale, évaluation

1. INTRODUCTION

L'apprentissage d'une langue est un long et laborieux processus. Il est tout de même possible grâce à l'enseignement de simplifier cette tâche. Cette approche se distingue par l'implication d'enseignants et l'utilisation d'outils, qu'ils soient numériques ou non, afin de guider les apprenants et de faciliter leur progression linguistique. Le but de ce projet est de réaliser un Proof Of Concept (POC) sur l'amélioration de l'apprentissage de l'anglais en recourant à des outils numériques de traitement du signal (MLSP) et de traitement du langage naturel (NLP). Ce faisant, il vise à corriger la prononciation et la grammaire des apprenants à travers la parole. La démographie initialement visée est constituée des individus dont la langue maternelle est l'indien et qui aspirent à maîtriser l'anglais. Pour mener à bien ce projet, il

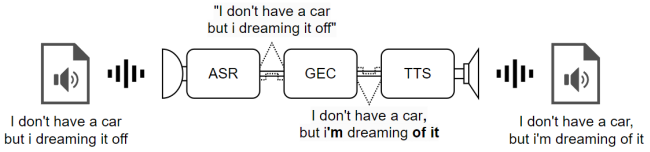

Fig. 1. Schéma de fonctionnement du pipeline

est envisageable de mettre en place un processus (pipeline) de plusieurs étapes (figure 1) :

1. Automatic Speech Recognition (ASR) : Retranscrire précisément les paroles de l'apprenant en un texte à l'aide d'un modèle Speech-To-Text (STT). Il est essentiel que ce modèle soit en capacité de nous fournir un texte fidèle incluant les erreurs grammaticales, et nous permettre un contrôle sur la correction de celles-ci.

2. Grammatical Error Correction (GEC) : Détecter et corriger des erreurs grammaticales du texte à l'aide d'un modèle Text-To-Text. Le modèle doit pouvoir nous proposer une ou plusieurs version corrigées, en prenant en compte le contexte intital.

3. Synthèse Vocale : Transformer le texte corrigé en un fichier audio à l'aide d'un modèle Text-To-Speech (TTS). Cela permettra à l'apprenant d'écouter la prononciation du texte corrigé.

Pour mettre en oeuvre ces étapes, nous comptons nous appuyer sur des modèles pré-existants et pré-entrainés. Cette approche présente plusieurs avantages avec en premier lieu des gains conséquents en termes de temps et d'argent, ainsi que l'utilisation de modèles déjà très performants. Nous envisageons de les perfectionner davantage en les affinants selon nos besoins :

— Pour le Speech-To-Text, l'affinage se concentrera sur les accents (anglais avec accent indien), ainsi que sur la précision et la fidélité de la transcription pour bien récupérer les possible erreurs grammaticales.

— Pour le Text-To-Text, l'affinage sera orienté sur la détection des erreurs grammaticales et leurs corrections pour avoir en sortie un modèle de correction des erreurs grammaticales (GEC).

Ce papier va être composé comme suit. On va en premier lieu réaliser une revue de la littérature pour situer et comparer notre projet avec les techniques existantes. On va par la suite étudier les différents jeux de données que nous utilisons pour affiner et évaluer nos modèles. Nous allons ensuite justifier l'utilisation de nos différents modèles au sein de notre pipeline, et finalement nous concluerons sur les biais et points forts de celle-ci notamment en répondant aux questions que nos évaluateurs nous ont posés.

2. ETAT DE L'ART

Notre POC contiendra donc les tâches suivantes : STT, GEC, TTS. Ce sont des tâches déjà utilisées dans divers contextes (assistants vocaux et virtuels, domaine de la santé, outils d'accessibilité, etc.). L'éducation émerge comme un domaine d'application en croissance, car la GEC permet de personnaliser l'apprentissage et de s'auto-corriger pour s'améliorer. Toutefois, les recherches existantes sur l'adaptation du STT et TTS sont éparpillées dans divers domaines d'application sans se concentrer particulièrement sur la pédagogie[1]. Nous allons donc organiser notre revue de littérature d'abord pour le domaine du STT, puis du TTS, et ensuite pour le GEC.

2.1. Speech-to-text

La conversion de la parole en texte et la reconnaissance automatique de la parole ont connu des avancées significatives ces dernières années, principalement grâce aux progrès fulgurants dans les domaines de l'apprentissage en profondeur et des approches basées sur les données. Le STT repose sur trois piliers majeurs : la modélisation acoustique, la modélisation linguistique et le décodage.

La modélisation acoustique a connu une évolution significative, passant des Modèles de Markov Cachés (HMM) traditionnels à l'utilisation de techniques d'apprentissage en profondeur telles que les Réseaux de Neurones Convolutifs (CNN), les Réseaux de Neurones Récursifs (RNN) et les modèles basés sur les Transformers. Ces avancées se sont traduites par des améliorations notables en termes de précision, notamment dans des environnements bruyants et pour des accents variés.

La modélisation linguistique, quant à elle, se concentre sur la compréhension du contexte linguistique des signaux vocaux. Si les modèles n-grammes traditionnels prédominaient, des avancées récentes ont exploré des approches plus avancées, comme les RNN, les réseaux LSTM (Long Short-Term Memory) et les modèles basés sur les Transformers. Ces avancées ont considérablement renforcé la capacité de gestion de la parole conversationnelle, des termes hors vocabulaire et du langage spécifique à des domaines particuliers.

Le décodage, troisième composante essentielle, consiste à traduire la sortie des modèles acoustiques et linguistiques en texte final. Les méthodes traditionnelles, comme le décodage de Viterbi basé sur les Modèles de Markov Cachés, ont été remplacées par les systèmes ASR de bout en bout, qui utilisent des techniques d'apprentissage en profondeur pour mapper directement la parole en texte. Ces systèmes ont démontré des résultats prometteurs en termes de précision et d'efficacité.[2]

2.2. Text-to-speech

En ce qui concerne la synthèse vocale, on retrouve aussi différentes approches : le TTS concaténatif, la synthèse de formants, le TTS par sélection d'unités et le TTS par apprentissage profond

La première approche est le TTS concaténatif qui repose sur l'enregistrement préalable de la parole humaine, suivi de la concaténation de petites unités de parole pour générer une parole synthétisée. Cette méthode est réputée pour sa capacité à produire une parole de haute qualité et naturelle. Cependant, elle présente des inconvénients majeurs, notamment le besoin de disposer d'une grande quantité de données de parole enregistrée, ainsi que des limitations dans la génération de variations de ton et de hauteur.

Une autre approche traditionnelle, la synthèse de formants, consiste à modéliser le tractus vocal pour générer la parole en manipulant les fréquences des formants des sons de parole. Cette méthode offre un meilleur contrôle sur des aspects tels que la hauteur et le timbre. Toutefois, elle peut parfois manquer de naturel et de réalisme.

Le TTS par sélection d'unités est une approche hybride qui combine des techniques de concaténation et de synthèse de formants. Cette méthode implique la sélection et la concaténation de petites unités de parole préenregistrée, tout en appliquant des techniques de synthèse de formants pour ajuster la parole. Cette approche permet de générer une parole de haute qualité et naturelle.

Mais récemment, les travaux de recherche se concentrent sur le TTS basé sur l'apprentissage en profondeur. Elle fait appel, encore une fois, à des réseaux neuronaux tels que les RNNs, les CNNs et les modèles basés sur les transformers pour générer la parole. Ces modèles ont la capacité d'apprendre à partir de grandes quantités de données, leur permettant de produire une parole d'une grande précision, avec une sonorité naturelle et une grande expressivité.

2.3. Grammatical Error Correction

Enfin, l'augmentation du nombre d'apprenants de la langue anglaise met en évidence la nécessité de développer des ressources automatisées ayant un impact pédagogique plus étendu. Parmi ces ressources, les efforts de recherche dans le domaine de la correction automatique des erreurs grammaticales (GEC) occupent une place importante. Jusqu'à récemment, les approches basées sur des règles et les modèles de traduction automatique, tels que les modèles de traduction

statistique basée sur des phrases, ont été privilégiés dans ce domaine. Cependant, il existe un intérêt marqué pour le développement de modèles de réseaux neuronaux capables de surmonter les limitations de ces approches et d'améliorer les résultats.[3]

En somme, notre projet s'inscrit pleinement dans l'ère actuelle car nous avons choisi de mettre en œuvre une approche neuronale pour aborder nos trois principales tâches. Notre approche consiste à exploiter des modèles pré-entraînés, une pratique largement adoptée aujourd'hui, que nous allons ensuite affiner pour répondre à nos besoins spécifiques. Nous souhaitons traiter de manière spécifique l'un des défis des systèmes de STT : la reconnaissance des accents.

3. JEUX DE DONNÉES

Comme indiqué dans l'introduction, nous avons mis en place un pipeline entre nos différents modèles (voir figure 1). Pour pouvoir évaluer notre pipeline, nous avons en premier lieu défini les jeux de données que nous allions utiliser et de quelle manière :

— **Jeu de données auto-généré** : Après des recherches infructueuse pour trouver des jeux de données d'enregistrements en anglais contenant des erreurs grammaticales, il a été décidé d'en créer un nous même.

Nous avons donc décidé d'utiliser un outil de synthèse vocale (gTTS qui se base sur l'API de Google Traduction) pour nous permettre de générer des audios à partir de textes avec des fautes grammaticales. gTTS a été choisi sur le fait qu'il soit premièrement simple d'utilisation, il permet de choisir des accents, mais aussi car il propage les diverses fautes du texte à l'oral. On a donc créé un jeu de données de 1141 entrées, composé d'audio anglais (avec des erreurs grammaticales et l'accent indien), et la phrase erronée correspondante à l'écrit. Ce jeu de données est utilisé pour la tâche d'ASR.

— **JFLEG** [4] : Jeu de données contenant des phrases erronées et leurs corrections (Corpus GEC). Composé de plus de 1500 items (754 pour l'entrainement et 747 pour test). C'est aussi avec ces données que nous avons auto-générés nos audios érronés, en utilisant gTTS (Voir ci-dessus "Jeu de données auto-généré").

— **Librispeech** [5] : Jeu de données audio contenant plus de 1000h d'audiobook en anglais avec la transcription. Ce jeu de données est utilisé pour la tâche de TTS.

Toutes ces données nous sont utiles pour l'affinage des modèles pré-entrainés, mais aussi pour les tester.

4. MODÉLISATION ET OPTIMISATION DES MODÈLES

4.1. ASR

La première tâche de notre pipeline correspond à la conversion des signaux audios d'un utilisateur en texte grâce à un modèle d'ASR. Nous avons donc évalué différents modèles d'ASR grâce à notre jeu de données auto-généré (composé d'audios contenant des erreurs grammaticales et des phrases correspondantes) pour en choisir un qui réponde aux défis suivants : parvenir à transcrire les audios en conservant les potentielles erreurs audibles afin de permettre à l'utilisateur de visualiser ses erreurs dans le texte ; parvenir à obtenir un modèle adapté à l'accent indien que comportent nos enregsitrements. Nous avons fait l'hypothèse qu'un modèle comportant une couche de traitement de la langue risquerait de corriger les fautes directement à cette étape, de ce fait on a comparé deux modèles : Wav2Vec2 qui ne comporte pas de modèle de langue, et Whisper qui lui est basé sur un modèle de langue.

Pour comparer les deux modèles, nous avons utilisé comme métrique d'évaluation le WER (Word Error Rate). Les résultats montrent que le modèle Whisper parvient finalement à bien retranscrire les erreurs avec un WER de 7% contre 19% pour le modèle Wav2Vec2. On a donc sélectionné le modèle Whisper pour la suite de nos expérimentations, et nous l'avons affiné à l'accent indien qui est présent dans nos données. En faisant cela, on est parvenu à améliorer le WER de 7% à 5%. Nous étions limité en terme de mémoire, mais rien qu'en faisant 200 pas (4 epochs) on observe une baisse importante de la perte du modèle (figure 2).

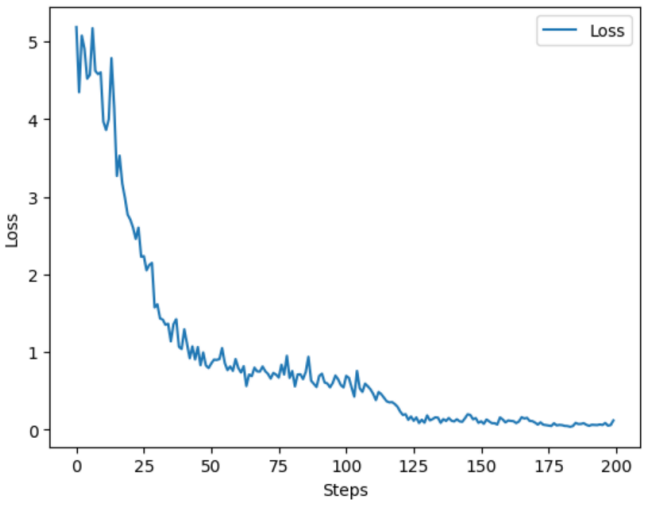

Fig. 2. Evolution de la perte du modèle Whisper durant le fine tunning

4.2. GEC

Ces dernières années, des tâches partagées telles que CoNLL-2014 [6] et BEA-2019 [7] ont été cruciales pour le développement de modèles de correction grammaticale. Bien que ces initiatives guident notre choix de modèles et de méthodes d'évaluation, il est important de noter que les avancées rapides de l'intelligence artificielle ont peut-être dépassé les modèles issus de ces tâches plus anciennes [8].

Nous nous sommes donc plutôt tournés vers les transformers, des modèles d'excellence pour la Correction Grammaticale (GEC) [9]. Parmi eux, le modèle **T5** de Google, un modèle encodeur-décodeur spécialisé dans les tâches textuelles, se distingue. Bien qu'il n'effectue pas directement la correction grammaticale, son utilisation est possible via le transfert d'apprentissage (3).

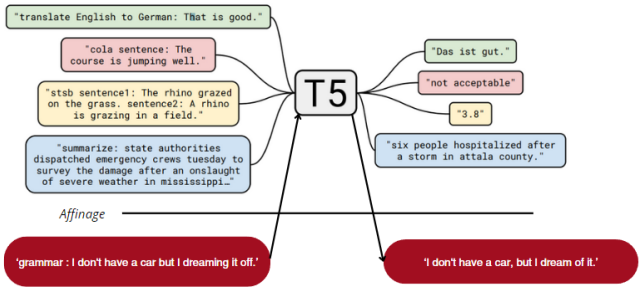

Fig. 3. Transfert d'apprentissage sur le modèle T5

L'évaluation d'un modèle de GEC est essentielle pour mesurer son efficacité, utilisant des métriques telles que la précision, le rappel et la F-mesure. Parmi les outils d'évaluation, ERRANT [10] et BERTScore [11] sont notables. ERRANT offre une évaluation sur les annotations, tandis que BERTScore utilise des embeddings contextualisés pour mesurer la similarité entre les séquences de mots. Ces outils fournissent des perspectives complémentaires à l'évaluation des modèles GEC. Nous utiliserons le dataset JFLEG [4] pour évaluer notre modèle GEC.

Méthode	Score
BERTScore - $f_{0.5}$-score	0.74 [0-1]
T5 - Sentence similarity	4.85 [0-5]
ERRANT - $f_{0.5}$-score	0.62 [0-1]

Table 1. Evaluation de T5

Un score de 0.62 pour ERRANT indique que le modèle fait des erreurs, mais la nature spécifique de ces erreurs peut varier (ponctuation, etc.). T5 nous montre une bonne similarité sémantique entre nos phrases, et BERTScore une bonne similarité sémantique entre les mots de nos phrases.

4.3. TTS

Nous avons utilisé le modèle gTTS [12] basé sur Google Translate, celui-ci est très simple à faire fonctionner et est suffisant dans notre cas. Comme autre modèle possible, il y a le modèle TacoTron2 [13]. Il prend en entrée un texte et génère un spectrogramme qui est ensuite passé dans un vocodeur pour générer le fichier audio. Grâce au spectrogramme généré, on peut espérer pouvoir évaluer la qualité du modèle durant l'évaluation.

Il est très difficile d'évaluer la qualité d'un modèle de TTS. En effet, il n'existe pas de métrique permettant de quantifier la qualité d'un fichier audio.

Nous utilisons principalement le MOS (Mean Opinion Score), une métrique subjective, en faisant écouter un fichier audio à un panel de personnes qui notent sa qualité de 1 à 5. Le MOS est ensuite calculé en prenant la moyenne des notes obtenues.

Nous nous basons sur le dataset LibriSpeech [5]. En générant des fichiers audio avec notre modèle TTS à partir des transcriptions, nous évaluons la qualité en attribuant un score d'opinion. Le MOS est calculé à partir de l'évaluation de deux évaluateurs, chacun notant 20 fichiers audio.

Méthode	Score
MOS	4.9 [1-5]

Table 2. TTS (gTTS) évaluation

Comme indiqué, nous avons tenté d'évaluer les audios en comparant les spectrogrammes (mel cepstral distortion) avec Tacotron2. Malheureusement, cette approche n'a pas pu être mise en œuvre à temps.

5. DISCUSSIONS

Notre travail contient des axes à améliorer que nous allons présenter dans cette section. Nous allons répondre notamment aux questions qui nous ont été posé par nos évaluateurs et qui touchent à trois thèmes différents : la diversité des accents, la diversité des langues, et la qualité des audios d'entrée.

5.1. Diversité des accents

Dans le cadre de notre projet, nous nous sommes limités au traitement d'un seul accent (en l'occurence celui indien) pour la tâche d'ASR. Ainsi, le modèle que nous avons affiné n'est en l'état pas adapté à gérer d'autres accents en entrée. Néanmois, étant donné que nous nous sommes basé sur le modèle Whisper, qui est un modèle très performant et ayant été entrainé lui même sur une diversité de langue et d'accent, il est possible que notre modèle parvienne tout de même à des résultats correctes avec un accent autre que celui indien. Néanmoins, un axe d'amélioration future serait d'avoir des

audios avec plusieurs accents et d'affiner et évaluer Whisper dessus. Mais cela cause de nouveaux le problème initial auquel on a fait face : le manque de données. On a été en mesure de créer 1141 audios, ce qui reste peu pour faire un affinnage et une évaluation d'un modèle, d'autant plus si on doit y insérer plusieurs accent. On se retrouverait alors avec seulement une centaine de données par accent et au plus on ajouterait d'accent au moins on aurait de données représentatrice d'un accent. En somme, si l'on souhaite affiné Whisper sur différents accents, nous aurons besoin de générer des audios qui en contiennent plusieurs. Nous aurions pour cela besoin de trouver un autre jeu de données à erreurs grammaticales sur lequel nous basé pour la génération d'audio, car celui de JFLEG ne contient pas plus de 2000 phrases erronées.

5.2. Diversité des langues

Tout comme nous nous sommes focalisés sur un seul accent pour ce travail, nous nous sommes également focalisés sur une seule langue : l'anglais. Cependant, les modèles que nous avons utilisé (WHisper, T5 et gTTS) sont des modèles pré-entrainés sur une multitude de langue (français, allemand, etc.). Il suffirait donc de refaire le même affinage que nous avons déjà fait sur les modèles, mais avec des données contenant des langues diverses Ainsi nous garderions trois modèles mais qui seraient multilingues. Encore une fois, le défi revient à trouver des données comportant des erreurs grammaticales à l'oral et dans des langues variées.

5.3. Qualité des audios d'entrée

Le modèle d'ASR Whisper a été entrainé sur des voix de femmes et des voix d'hommes, mais pas avec des voix d'enfants. Certains modèles existent et sont spécialement affinés pour les voix d'enfants [14] donc c'est possible d'améliorer les performances du modèle sur ce cas là grâce à des données contenant des audios d'enfants. Mais dans notre cas, le modèle de TTS pour générer nos données audios ne possèdait pas d'option pour le choix de la voix. Ensuite, des erreurs de prononciation ou de grammaire peuvent être beaucoup plus fréquentes chez les enfants qui apprennent une langue étrangère. Notre outils est fait pour détecter les erreurs qui soient audibles, mais il faut que la phrase ait un minimum de contexte (donc de mots correctes) pour que le modèle de GEC parvienne à comprendre le contexte et ainsi corrigé correctement les mots erronés. Nous recommandons plutôt d'utiliser notre outils avec des personnes (enfants ou adultes) qui ont un minimum de connaissances dans la langue cible pour construire des phrases sémantiquement correctes.

De plus, nous avons dû générer nous mêmes notre jeu de données d'entrée (génération des audios à partir de phrases erronées). Etant donné que c'est généré artificiellement, les audios ne contiennent pas de bruit en fond. Pou rendre notre modèle d'ASR plus robuste, on pourrait essayer d'y ajouter du bruit manuellement par exemple en couplant les spectrogrammes de nos audios avec ceux de bruit de fond comme de la musique, des conversations, etc.

5.4. Biais du pipeline

Nous avons réalisé des tests de notre pipeline en nous enregistrant en train de dire différentes phrases en anglais et comportant différents types d'erreurs. On a aussi réalisé un enregistrement avec de la musique en fond pour tester notamment la partie speech-to-text. Ces tests sont disponibles sur notre répertoire github (lien dans l'entête de ce papier). Ils ont révélé que le pipeline fonctionne bien, et qu'il arrive à extraire et corriger la majorité des fautes. L'ASR restranscris fidèlement les phrases, mais il faut faire attention a bien articuler. Quant au GEC il reste une part d'imprécision sur la correction qui amène à des phrases mal formulées.

6. CONCLUSION

En conlusion, l'idée de notre projet est de développer un outils d'aide à l'apprentissage d'une langue étrangère. Cela est parti d'un constat personnel que nous apprenons mieux une langue lorsqu'on la pratique à l'oral. De ce fait, nous avons développé un POC constitué de trois modèles : une tâche d'ASR, une tâche de GEC et une tâche de TTS. Dans l'ensemble nos modèles ont obtenu de bonne performance et ont montré une bonne capacité à s'adapter à nos données audio qui comportent des erreurs grammaticales et un accent indien. Des points restent à améliorer pour obtenir un outils plus général et plus robuste, en particulier la diversité de nos données d'entrée du pipeline. Nous avons été fortement limité par le manque de données open source contenant des audios avec des erreurs audibles et le texte retranscrit. Nous les avons donc générer, mais étions contraint au limite du modèle TTS qu'on a utilisé : manque de divesrité dans les voix disponible, manque de diversité pour les accents disponibles, et limite dans le nombre de requête API envoyée au modèle (création de seulement 1141 audios).

7. REFERENCES

[1] Daniel Markgraf Susanne Robra-Bissantz Ricarda Schlimbach, Heidi Rinn, "A litterature review on pedagogical conversationnal agent adaptation," 2022.

[2] V. Madhusudhana Reddy, T. Vaishnavi, and K. Pavan Kumar, "Speech-to-text and text-to-speech recognition using deep learning," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 2023, pp. 657–666.

[3] Ayush Gupta Yokila Arora, Jaspreet Kaur, "Grammaticalerrorcorrectionusingneuralnetworks," .

[4] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault, "Jfleg: A fluency corpus and benchmark for grammatical error correction," 2017.

[5] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[6] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, "The CoNLL-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, Eds., Baltimore, Maryland, June 2014, pp. 1–14, Association for Computational Linguistics.

[7] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, Eds., Florence, Italy, Aug. 2019, pp. 52–75, Association for Computational Linguistics.

[8] Muhammad Reza Qorib, Seung-Hoon Na, and Hwee Tou Ng, "Frustratingly easy system combination for grammatical error correction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, Eds., Seattle, United States, July 2022, pp. 1964–1974, Association for Computational Linguistics.

[9] Muhammad Reza Qorib and Hwee Tou Ng, "Grammatical error correction: Are we there yet?," in *Proceedings of the 29th International Conference on Computational Linguistics*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, Eds., Gyeongju, Republic of Korea, Oct. 2022, pp. 2794–2800, International Committee on Computational Linguistics.

[10] Christopher Bryant, Mariano Felice, and Ted Briscoe, "Automatic annotation and evaluation of error types for grammatical error correction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan, Eds., Vancouver, Canada, July 2017, pp. 793–805, Association for Computational Linguistics.

[11] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, "Bertscore: Evaluating text generation with bert," 2020.

[12] "gTTS — gTTS documentation," .

[13] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2018.

[14] Mariam Yiwere Peter Corcoran-Horia Cucu Rishabh Jain, Andrei Barcovschi, "Adaptation of whisper models to child speech recognition," .