

DÉPARTEMENT INFORMATIQUE - IUT 2 GRENOBLE



Année Universitaire 2023-2024

MÉMOIRE DE STAGE

**Groupe d'Étude pour la Traduction Automatique et le Traitement
Automatisé des Langues et de la Parole**

Laboratoire d'Informatique de Grenoble



Présenté par

Enzo Vigne

Jury

IUT : M. Franck Corset

IUT : Mme. Sophie Dupuy-Chessa

Société : Mme. Cécile Macaire

Déclaration de respect des droits d'auteurs

Par la présente, je déclare être le seul auteur de ce rapport et assure qu'aucune autre ressource que celles indiquées n'ont été utilisées pour la réalisation de ce travail. Tout emprunt (citation ou référence) littéral ou non à des documents publiés ou inédits est référencé comme tel et tout usage à un outil doté d'[IA⁴⁷⁸⁴⁷](#) a été mentionné et sera de ma responsabilité.

Je suis informé qu'en cas de flagrant délit de fraude, les sanctions prévues dans le règlement des études en cas de fraude aux examens par application du décret 92-657 du 13 juillet 1992 peuvent s'appliquer. Elles seront décidées par la commission disciplinaire de l'UGA.

A, Grenoble,

Le, 09/06/2024

Signature :

REMERCIEMENTS

Tout d'abord, je tiens à remercier toute l'équipe de recherche et développement, nommée Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues de la Parole (GETALP), du Laboratoire Informatique de Grenoble (LIG) que j'ai intégré, pour leur accueil, leur soutien et la disponibilité dont ils ont fait preuve pour me venir en aide au besoin.

De plus, je tiens particulièrement à remercier :

Mme. Macaire, pour m'avoir accordé sa confiance et attribué une mission intéressante, valorisante et riche en apprentissage, ainsi que pour sa précieuse aide lors de la rédaction du rapport.

M. Arrigo, pour l'aide qu'il a su m'apporter tout au long du projet, en particulier pour ses réponses aux problèmes techniques auxquels j'ai pu faire face.

M. Schwab, pour sa disponibilité et sa capacité à venir en aide.

Faire mon stage parmi eux a été un plaisir, autant sur le plan professionnel que personnel, j'ai pu beaucoup apprendre, approfondir mes connaissances sur différentes technologies en plus du soutien vis-à-vis de mon projet professionnel.

Enfin, je tiens aussi à remercier M. Corset, mon tuteur pédagogique, pour sa disponibilité en cas de besoin et sa bienveillance, ainsi que M. Goulian, qui a fourni une grande aide dans le processus de recherche de stage.

Tables des matières

I. INTRODUCTION	3
II. PRÉSENTATION DU CADRE GÉNÉRAL	4
II.1 Contexte	4
II.2 Objectifs du projet	4
II.3 Technologies utilisées	5
Angular	5
Javascript	6
Python	7
GitHub	8
II.4 Analyse de l'existant	8
Page Picto	9
Page d'authentification	12
Page PostEditPicto	13
Page DicoPicto	14
II.5 Impact environnemental et sociétal	14
III. RÉALISATIONS	15
III.1 Gestion de projet	16
Fonctionnement global au sein de l'équipe	16
Organisation du travail	16
III.2 Développement des fonctionnalités.	16
Page SpeechToPicto	16
Front-End de la page	18
MediaRecorder	22
Intégration des scripts Python	24
En cours	24
Travail restant	25
CONCLUSION	27
GLOSSAIRE	28
ANNEXES	31
RÉFÉRENCES BIBLIOGRAPHIQUE ET SITOGRAPHIQUE	34

Table des figures

Figure 1 : Page Picto du site InterAActionPicto - Requête.....	9
Figure 2 : Page Picto du site InterAActionPicto - Personnalisation Sortie.....	10
Figure 3 : Page Picto du site InterAActionPicto - Export PDF.....	12
Figure 4 : Page d'Authentification.....	13
Figure 5 : Page de post édition des pictogrammes.....	13
Figure 6 : Page de dictionnaire de pictogramme.....	14
Figure 7 : Maquette de l'interface de la page SpeechToPicto.....	18
Figure 8 : Page SpeechToPicto.....	19
Figure 9 : Traduction à partir d'un texte.....	19

Figure 10 : Traduction à partir de phrases prédéfinies.....	20
Figure 11 : Traduction à partir d'un fichier audio.....	20
Figure 12 : Traduction à partir de la voix.....	21
Figure 13 : Code MediaRecorder.....	23

I. INTRODUCTION

En tant qu'étudiant en troisième année de BUT (Bachelor Universitaire Technologique) d'informatique au sein de l'IUT2 de Grenoble. J'ai eu la chance de pouvoir effectuer un stage dans le cadre de ma formation et de la validation de mon année. La période pendant laquelle ce stage s'est déroulé est du 17 avril 2023 au 23 juin 2023.

Le stage s'est produit au sein du Laboratoire Informatique de Grenoble (LIG), créé le premier janvier 2007, il est aujourd'hui dirigé par Noël DE PALMA, il est possible de consulter [l'organigramme du LIG](#) en annexe (mis à jour pour la dernière fois en septembre 2023). Le laboratoire est en partenariat avec diverses organisations telles que l'Université Grenoble Alpes (UGA), l'Institut National de Recherche en Informatique et en Automatique (INRIA), le Centre National de Recherche Scientifique (CNRS) ainsi que l'Institut Polytechnique de Grenoble (INP). Le laboratoire compte un total de 22 équipes qui se basent sur différents axes. L'équipe GETALP, que j'ai intégrée, mène des recherches sur la thématique de Systèmes intelligents pour les données.

Plus précisément, GETALP participe à divers projets, ma mission porte sur le projet nommé InterAActionPicto. Il s'agit d'une plateforme permettant aux personnes en situation de [handicap langagier](#)¹ de communiquer par le biais de suites de [pictogrammes](#)² traduisant une phrase saisie par l'utilisateur. Les requêtes des utilisateurs peuvent, s'ils le souhaitent, être enregistrées afin d'en tirer des données sur la satisfaction concernant leur utilisation de l'outil.

Le sujet de mon stage consiste à intégrer dans ce projet des fonctionnalités permettant une reconnaissance plus précise des requêtes des utilisateurs, et ce, sous différents formats (audio ou textuels). Cela est possible grâce à l'utilisation du modèle de [Reconnaissance Automatique de la Parole \(RAP\)](#)³ nommé Whisper, développé par l'entreprise OpenAI. Ce modèle fonctionne sur la base de l'[Intelligence Artificielle \(IA\)](#)⁴.

Les technologies utilisées dans le cadre de ce projet sont [Angular](#)⁵, un [framework](#)⁶ basé sur le langage [TypeScript](#)⁷, [Javascript](#)⁸ ainsi que [Python](#)⁹. Ce rapport présentera de manière plus complète l'existant, les technologies utilisées ainsi que les objectifs de ma mission.

Je vais dans un premier temps détailler davantage le contexte du stage, ses objectifs, ce qui existe déjà, les technologies utilisées ainsi que l'impact environnemental global du stage. Dans un deuxième temps, je présenterai la gestion du projet, ce que j'ai réalisé, les difficultés que j'ai pu rencontrer ainsi que le travail qu'il reste à faire.

II. PRÉSENTATION DU CADRE GÉNÉRAL

La présentation du cadre général du stage vise à fournir des précisions détaillées sur le fonctionnement et l'organisation du stage, ainsi que sur le contexte spécifique dans lequel il se déroule. Cette présentation a pour but de clarifier les objectifs du stage. Elle inclut également une description des technologies et des outils qui seront utilisés pendant la période de stage. Par ailleurs, cette présentation sert à situer le stage dans son cadre plus large, en expliquant son importance et sa pertinence par rapport aux projets en cours ou aux missions de l'organisation.

II.1 Contexte

Le projet InterAActionPicto s'inscrit dans une volonté de permettre aux personnes qui ne savent pas utiliser les pictogrammes de communiquer avec une personne en situation de handicap langagier. Il s'agit d'une initiative visant à promouvoir l'inclusivité et l'accessibilité pour les personnes ayant des difficultés de communication. Actuellement, les personnes en situation de handicap utilisent des outils de Communication Alternative et Augmentée (CAA) pour interagir avec leur entourage. La CAA regroupe divers moyens et stratégies permettant de compenser les difficultés de communication, que ce soit par des gestes, des symboles, des pictogrammes ou des technologies avancées.

Ces outils de CAA, bien qu'efficaces, sont souvent coûteux et demandent une longue période d'apprentissage. La complexité de leur utilisation peut représenter une barrière significative, tant pour les utilisateurs directs que pour leur entourage. Proposer un outil capable de traduire automatiquement le texte ou l'oral en pictogrammes permettrait de surmonter ces obstacles et de faciliter la communication entre une personne qui ne maîtrise pas ces outils et une personne en situation de handicap langagier.

Dans le cadre du projet Propicto ([voir site Propicto](#)), des modèles ont été développés à cet effet. Ces modèles reposent sur des systèmes d'Intelligence Artificielle (IA) et de [Traitement du Langage Naturel \(NLP\)](#)¹⁰. Cependant, leur utilisation nécessite des compétences techniques, notamment en programmation et en Python.

Ainsi, le projet InterAActionPicto vise non seulement à faciliter la communication pour les personnes en situation de handicap langagier, mais aussi à renforcer l'accessibilité en réduisant les barrières technologiques et cognitives existantes.

II.2 Objectifs du projet

Des outils permettant de traduire automatiquement la parole et le texte en pictogrammes Arasaac ont été développés. Cependant, ces outils ne sont pas encore disponibles pour un usage grand public, ce qui souligne la nécessité de créer une plateforme web dédiée. L'objectif du projet InterAActionPicto, comme mentionné précédemment, est de

permettre aux personnes en situation de handicap langagier de communiquer par le biais de pictogrammes.

Plus précisément, l'utilisateur pourra traduire ses mots en une suite de pictogrammes (dessins schématiques) provenant de l'[API](#)¹¹ Arasaac. La mission de mon stage consiste à intégrer ces pictogrammes directement depuis l'API, plutôt que de les récupérer d'une liste prédéfinie stockée dans une base de données. Cette approche permettra de garantir la mise à jour constante des pictogrammes disponibles et d'améliorer la flexibilité de l'outil.

De plus, un des objectifs principaux de mon stage est de mettre en place un système de Reconnaissance Automatique de la Parole (RAP). La RAP consiste à convertir la parole de l'utilisateur en texte, qui sera ensuite traduit en une série de pictogrammes à l'aide de l'API Arasaac. Ce système permettra aux utilisateurs de communiquer plus facilement et naturellement, en parlant directement à l'outil qui traduira leurs paroles en pictogrammes compréhensibles pour les personnes en situation de handicap langagier.

En résumé, les principaux objectifs de mon stage sont :

- Intégrer les pictogrammes de l'API Arasaac dans la plateforme web.
- Développer et implémenter un système de Reconnaissance Automatique de la Parole pour convertir la parole en texte.
- Traduire le texte obtenu en une suite de pictogrammes grâce à l'API Arasaac.

Ces objectifs visent à améliorer l'accessibilité et l'efficacité des outils de communication pour les personnes en situation de handicap langagier, tout en recueillant des retours d'utilisateurs pour affiner et optimiser la solution proposée.

II.3 Technologies utilisées

Angular

Le projet InterAActionPicto utilise principalement Angular, un framework de développement web maintenu par Google. Angular est largement utilisé pour créer des applications web dynamiques et robustes, et il s'est avéré être un choix idéal pour notre projet.

Angular, dans sa version moderne, repose sur TypeScript, un sur-ensemble de JavaScript qui offre une syntaxe typée statiquement, c'est-à-dire qu'il propose une syntaxe plus riche, mais le langage est ensuite transformé en JavaScript plutôt qu'interprété. Cette particularité améliore la qualité du code en facilitant la détection des erreurs dès la compilation. De plus, TypeScript permet une meilleure autocomplétion et une navigation optimisée dans le code grâce aux [IDE](#)¹² modernes, rendant le développement plus fluide et productif.

L'un des aspects les plus avantageux d'Angular est son architecture basée sur des [composants](#)¹³. Chaque composant est une entité autonome qui encapsule sa logique et son

interface utilisateur (template). Cette modularité favorise la réutilisation du code et simplifie la maintenance de l'application.

Angular excelle également par son mécanisme de data binding bidirectionnel. Cette fonctionnalité synchronise automatiquement les données entre le modèle et la vue, réduisant considérablement la quantité de code nécessaire pour gérer les interactions utilisateur et les mises à jour de l'interface. Par exemple, lorsque l'utilisateur modifie un formulaire, les données associées sont instantanément mises à jour.

Le routage intégré d'Angular est aussi un outil puissant pour la gestion de la navigation dans les applications. Grâce au module de routage, il est possible de configurer facilement des chemins dynamiques qui chargent les composants appropriés en fonction de l'[URL](#)¹⁴. Il s'agit d'une fonctionnalité particulièrement utile pour créer un [Single Page Application \(SPA\)](#)¹⁵ où les utilisateurs peuvent naviguer entre différentes sections sans recharger la page entière.

Enfin, Angular encourage les bonnes pratiques de développement avec ses outils intégrés pour les tests unitaires et les tests de bout en bout.

En résumé, l'utilisation d'Angular dans notre projet offre de nombreux avantages, allant de la modularité et la réutilisabilité du code à une meilleure gestion des données et une navigation fluide. Ces caractéristiques facilitent non seulement le développement, mais assurent aussi que notre application soit maintenable et évolutive à long terme. Travailler avec Angular m'a permis d'apprécier la puissance et la flexibilité de ce framework, enrichissant ainsi mon expérience et mes compétences en développement web.

Javascript

Du code JavaScript soit utilisé dans le projet InterAAActionPicto, il joue un rôle crucial dans certaines fonctionnalités spécifiques. L'une de ces fonctionnalités est la [lemmatisation](#)¹⁶, un processus linguistique essentiel pour améliorer la précision de la traduction en pictogrammes.

La lemmatisation est une technique de traitement du langage naturel (NLP) qui consiste à réduire les mots à leur forme de base, appelée "lemme". Par exemple, différentes formes d'un verbe comme "courir", "courait" et "courraient" sont toutes ramenées à leur lemme commun "courir". Ce processus permet de standardiser les mots avant leur traitement ultérieur, ce qui est particulièrement utile pour des applications telles que la traduction en pictogrammes.

Dans le cadre de notre projet, le JavaScript est utilisé pour implémenter la lemmatisation directement au sein de la plateforme web. Cela permet de traiter le texte entré par les utilisateurs de manière plus efficace et précise. En normalisant les mots avant de les envoyer à l'API Arasaac pour traduction en pictogrammes, nous pouvons garantir que les pictogrammes générés sont les plus pertinents possibles, indépendamment de la forme initiale du mot.

L'intégration de la lemmatisation via JavaScript améliore ainsi la qualité de la communication assistée par pictogrammes. Elle permet de s'assurer que des variantes de mots, qu'elles soient des conjugaisons ou des déclinaisons, ne perturbent pas le processus de traduction et que les utilisateurs reçoivent des représentations pictographiques précises et cohérentes.

En somme, même si le JavaScript n'est pas omniprésent dans le projet, son utilisation pour la lemmatisation est une composante clé qui contribue de manière significative à l'efficacité et à la précision de l'outil de traduction en pictogrammes d'InterAActionPicto.

Python

Le projet InterAActionPicto intègre également du code en langage Python, un choix motivé par les nombreuses qualités de ce langage. Python est reconnu pour sa simplicité et sa lisibilité, ce qui permet d'accélérer le développement et de faciliter la maintenance du code. De plus, il bénéficie d'une vaste bibliothèque standard ainsi que de nombreux modules externes, qui simplifient l'intégration de fonctionnalités complexes sans nécessiter un développement entièrement personnalisé.

Au cours de ce stage, plusieurs scripts en Python jouent un rôle essentiel dans le [traitement automatique des langues \(TAL\)](#)¹⁷. Un des outils clés que nous utilisons est Whisper, un modèle avancé de reconnaissance vocale. Whisper est capable de transcrire des fichiers audio en texte de manière efficace. Bien que Whisper supporte plusieurs langues, notre projet se concentre spécifiquement sur la retranscription des audios en français. Cette transcription est ensuite utilisée pour générer les pictogrammes correspondants via l'API Arasaac.

En outre, d'autres scripts Python sont employés pour nettoyer et affiner la sortie générée par Whisper. Ce processus de nettoyage est crucial pour garantir que le texte obtenu est précis et exempt d'erreurs qui pourraient affecter la qualité de la traduction en pictogrammes. Par exemple, ces scripts peuvent corriger des erreurs de transcription, supprimer des bruits de fond ou reformater le texte pour qu'il soit compatible avec les exigences de l'API Arasaac.

L'utilisation de Python dans le projet permet de tirer parti de puissants outils de traitement du langage naturel et de reconnaissance vocale, tout en assurant une flexibilité et une efficacité accrues dans le développement et la maintenance des fonctionnalités. Grâce à Python, nous pouvons intégrer des technologies avancées comme Whisper pour améliorer la qualité de notre solution de communication par pictogrammes, rendant ainsi notre plateforme plus fiable et accessible pour les utilisateurs.

En somme, Python joue un rôle indispensable dans le projet InterAActionPicto, apportant des solutions robustes pour le traitement automatique des langues et la reconnaissance vocale, ce qui contribue significativement à l'amélioration de l'expérience utilisateur et à l'atteinte des objectifs du projet.

GitHub

Durant mon stage, j'ai également eu l'occasion de travailler avec GitHub, une plateforme de gestion de versions et de collaboration de code extrêmement populaire dans le développement logiciel. GitHub a été un outil central dans notre flux de travail, facilitant la gestion du code, la collaboration et l'intégration continue.

Au début de mon stage, j'ai appris à utiliser le concept de "fork" sur GitHub. Forker un projet signifie créer une copie personnelle du dépôt de code original. Cette approche m'a permis de travailler sur le code de manière indépendante sans affecter le projet principal. Après avoir fait un fork du projet d'origine, j'ai pu cloner le dépôt sur ma machine locale pour commencer à travailler.

Pour gérer les différentes branches du projet, effectuer des commits et des pushes, j'ai utilisé GitHub Desktop, un outil que j'ai découvert et appris à utiliser pendant mon stage. GitHub Desktop est une application qui offre une interface graphique pour interagir avec GitHub, rendant la gestion de version plus accessible, surtout pour ceux qui préfèrent une interface utilisateur intuitive plutôt que la ligne de commande.

Grâce à GitHub Desktop, j'ai pu créer et basculer facilement entre les branches. La gestion des branches est essentielle dans un projet de développement, car elle permet de travailler sur de nouvelles fonctionnalités ou des correctifs de manière isolée avant de les fusionner avec la branche principale. J'ai également appris à effectuer des commits pour sauvegarder des instantanés de mon travail, et à pousser ces commits vers mon dépôt distant sur GitHub.

En outre, GitHub offre des fonctionnalités puissantes pour la gestion des projets et des équipes, notamment le système d'issues qui permet d'établir les tâches à accomplir de manière claire et efficace.

En résumé, l'utilisation de GitHub pendant mon stage a été extrêmement enrichissante. J'ai pu gérer efficacement mon code, utiliser des outils comme GitHub Desktop pour simplifier et optimiser mon flux de travail. GitHub a non seulement facilité la gestion du code, mais par ailleurs a facilité la collaboration autour de mon code, particulièrement lorsque je rencontrais des problèmes face auxquels j'avais besoin d'aide. Cette expérience m'a permis de développer des compétences pratiques et précieuses en gestion de versions et en collaboration de code, renforçant ainsi mes capacités en tant que développeur.

II.4 Analyse de l'existant

Le projet étudié dans ce rapport est déjà en place et a démontré son utilité dans son domaine d'application. Cette section a pour objectif de présenter en détail l'existant, en décrivant les principales caractéristiques, fonctionnalités déjà présentes dans le projet. Cette analyse permettra de mieux comprendre les fondements du projet, d'identifier les éléments qui fonctionnent efficacement et de repérer les aspects nécessitant des améliorations ou des

évolutions. Je vais donc ici présenter les pages les plus importantes déjà implémentées dans le projet, en détailler les fonctionnalités et l'utilité.

Les fonctionnalités qui seront présentées sont celles qui sont trouvables sur le [site de l'application](#) hébergée dans les serveurs du bâtiment IMAG (bâtiment dans lequel l'équipe GETALP se trouve).

Page Picto

L'une des pages principales de la plateforme est celle destinée aux utilisateurs souhaitant exploiter ses fonctionnalités. Cette page permet à un utilisateur de formuler des requêtes pour traduire des phrases en pictogrammes. La capture d'écran ci-contre illustre cette page en détail.

Choisir la/les banque(s) d'images

- ☒ Arasaac
☒ Mulberry
☒ Lemmatisation

Entrez un mot à rechercher

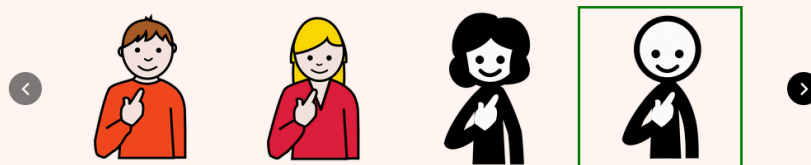
Bonjour je m'appelle Enzo Vigne

☐ no Je veux aider la recherche et j'accepte que mes données soient traitées, utilisées voire diffusées de manière anonymes

Résultat pour le mot : "Bonjour"



Résultat pour le mot : "je"



Interaaction Picto 1.0.0

Figure 1 : Page Picto du site InterAActionPicto - Requête

Je vais maintenant fournir des explications plus précises sur le fonctionnement de cette plateforme.

Tout d'abord, l'utilisateur saisit la phrase qu'il souhaite traduire sous forme de pictogrammes dans la zone de texte. Pour des raisons de temps de réponse, un maximum de cinquante mots est accepté lors de la saisie. L'utilisateur a la possibilité d'activer ou de désactiver certaines options. Parmi ces options, la lemmatisation permet de simplifier les formes des mots pour une meilleure correspondance avec les pictogrammes. La lemmatisation est le processus de réduction des mots à leur forme de base ou lemme ([voir détails sur la lemmatisation](#)).

En outre, l'utilisateur peut choisir d'inclure des banques d'images spécifiques telles qu'Arasaac et Mulberry, offrant ainsi une plus grande variété de pictogrammes pour chaque mot ou expression. Il est important de noter que dans cette page, les pictogrammes, bien que trouvables dans les banques d'images Arasaac et Mulberry, sont stockées sur le serveur qui héberge la plateforme (il n'y a donc pas de communication à l'API d'Arasaac).

Sous la zone de saisie, une option permet également de traiter, utiliser ou diffuser les données de manière anonyme, garantissant ainsi la confidentialité des informations fournies par l'utilisateur.

Une fois la requête validée, la plateforme affiche des images correspondant à chaque mot de la phrase saisie. L'utilisateur peut alors sélectionner les pictogrammes qu'il juge les plus appropriés, offrant ainsi une flexibilité et une personnalisation accrues dans la traduction visuelle.

Cette interface permet à la plateforme de répondre efficacement aux besoins des utilisateurs en matière de communication visuelle, grâce à l'utilisation de banques d'images reconnues.

Police

Taille de police 16

Type de police

type de police
Arial

Couleur de police

Transformation

☒ Aucun ☐ Minuscule ☐ Majuscule ☐ Capitale

Bordure

Type de bordure

type de bordure

Taille de bordure 5

Couleur de bordure

Ecriture dans le pictogramme

☒ Mots dans les pictogrammes ☐ Mots en dehors des pictogrammes

☒ Mots en bas du pictogramme ☐ Mots en haut du pictogramme

Résultat de la sélection

InterAction Picto 1.0.0

Figure 2 : Page Picto du site InterAActionPicto - Personnalisation Sortie

Une fois la sélection des pictogrammes terminée, l'utilisateur a ensuite la possibilité de personnaliser la sortie des pictogrammes grâce à un menu dédié, offrant diverses options pour adapter l'apparence des pictogrammes selon ses préférences personnelles. Cette personnalisation inclut des modifications de la police et des bordures des pictogrammes, ainsi que le positionnement du texte qui les accompagne.

Pour ce qui est de la police, l'utilisateur peut effectuer plusieurs ajustements :

- **Taille de la police** : L'utilisateur peut ajuster la taille du texte qui accompagne chaque pictogramme, permettant ainsi d'adapter la lisibilité en fonction des besoins spécifiques.
- **Type de police** : Il est possible de choisir parmi plusieurs types de polices disponibles, telles qu'Arial, Karumbi et Chilanka, afin de s'assurer que le texte est esthétiquement plaisant et facile à lire.
- **Transformation du texte** : L'utilisateur peut également choisir la transformation du texte, en sélectionnant entre aucune transformation, en minuscule, en majuscule ou en capitale, ce qui permet d'uniformiser l'apparence du texte selon les préférences ou les normes de style.

En ce qui concerne les bordures, plusieurs options de personnalisation sont disponibles :

- **Type de bordure** : L'utilisateur peut sélectionner le style de la bordure entourant chaque pictogramme, avec des options telles que le trait continu ou les pointillés, afin de mieux structurer l'affichage visuel.
- **Taille de la bordure** : Il est possible d'ajuster l'épaisseur de la bordure, offrant une flexibilité supplémentaire pour améliorer la clarté et l'impact visuel des pictogrammes.

Enfin, le positionnement du texte accompagnant les pictogrammes peut être configuré selon les préférences de l'utilisateur :

- **Emplacement du texte** : L'utilisateur peut choisir de placer le texte à l'extérieur ou à l'intérieur du pictogramme, ce qui permet d'optimiser l'utilisation de l'espace et de mieux intégrer le texte avec les images.
- **Position du texte** : Le texte peut être positionné sous ou au-dessus du pictogramme, offrant ainsi une autre dimension de personnalisation pour assurer que le message est clair et bien organisé.

Ces options de personnalisation permettent à l'utilisateur de créer une sortie de pictogrammes qui est non seulement fonctionnelle, mais aussi visuellement attrayante et adaptée à des besoins spécifiques, améliorant ainsi l'efficacité de la communication visuelle.

Résultat de la sélection

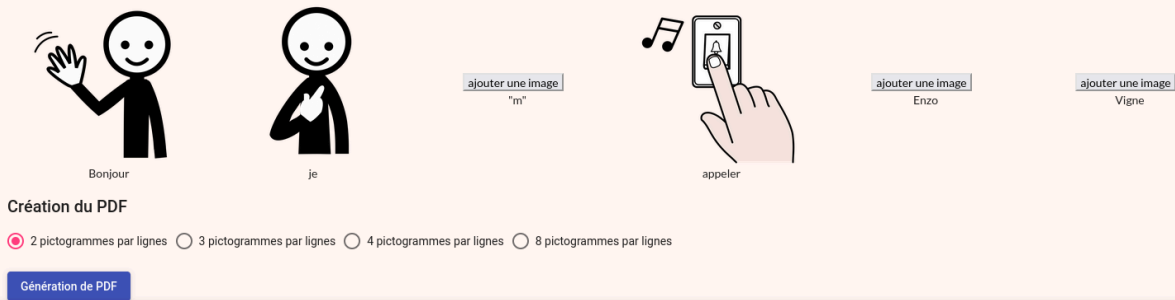


Figure 3 : Page Picto du site InterAActionPicto - Export PDF

Enfin, une fois ces étapes de personnalisation terminées, l'utilisateur peut visualiser un aperçu du résultat de sa sélection en bas de la page. Cet aperçu permet de vérifier que tous les pictogrammes et textes apparaissent comme souhaité avant de finaliser la sortie. Si certaines images n'ont pas été trouvées ou si les pictogrammes générés ne sont pas satisfaisants, l'utilisateur a la possibilité d'ajouter manuellement une image alternative pour remplacer le pictogramme manquant ou incorrect. Cette fonctionnalité assure que la traduction est complète et répond aux attentes spécifiques de l'utilisateur.

En plus de ces ajustements, l'utilisateur peut également choisir le nombre de pictogrammes par ligne dans le document PDF final. Cette option permet d'optimiser la présentation du document en fonction de la densité d'informations souhaitée, garantissant ainsi une lecture claire et structurée. Une fois tous les ajustements effectués et l'aperçu validé, il ne reste plus qu'à exporter le document finalisé sous forme de PDF.

Cette étape d'exportation génère un document prêt à l'emploi, intégrant toutes les personnalisations et les ajustements effectués par l'utilisateur, ce qui facilite le partage et l'utilisation des pictogrammes pour la communication visuelle.

Page d'authentification

The screenshot shows the 'Connexion' (Login) page. It has a light gray header with the title 'Connexion'. Below the header, there are two input fields: 'Nom de compte :' with the placeholder text 'Account', and 'Mot de passe :' with the placeholder text 'Password'. At the bottom, there is a blue 'Submit' button.

Figure 4 : Page d'Authentification

Le site InterAActionPicto dispose aussi d'une interface basique de connexion (visible ci-contre), elle n'est utilisée que pour les utilisateurs administrateurs pour accéder à différentes fonctionnalités que je détaillerai par la suite.

Page PostEditPicto

Cette page fournit des instructions détaillées que l'utilisateur doit suivre. Elle affiche des phrases accompagnées de leurs traductions en pictogrammes, avec pour objectif de valider, corriger ou améliorer ces traductions. L'utilisateur peut examiner chaque traduction et décider si elle est correcte ou nécessite des modifications. Pour cela, il a la possibilité de supprimer des pictogrammes existants ou d'en ajouter de nouveaux pour obtenir le résultat souhaité.

Pour ajouter un pictogramme, l'utilisateur utilise la même fonctionnalité de recherche de pictogrammes que celle précédemment présentée ([voir Picto](#)), ce qui lui permet de trouver rapidement et facilement les pictogrammes nécessaires. Une fois qu'il est satisfait de la traduction d'une phrase, il peut la valider et passer à la phrase suivante. Cette approche permet de s'assurer que chaque phrase est correctement traduite avant de continuer.

À tout moment, l'utilisateur peut choisir de quitter cette activité s'il le souhaite. Cette page est conçue pour faciliter le processus de validation et de correction des traductions, en offrant une interface et des outils efficaces pour améliorer la précision et la qualité des pictogrammes utilisés.

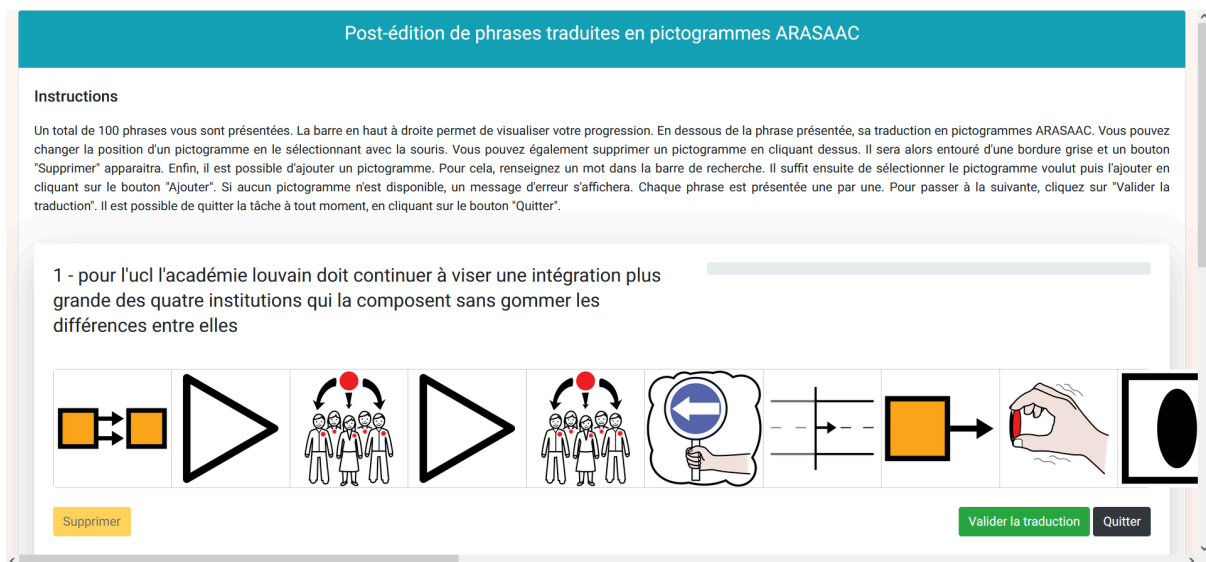


Figure 5 : Page de post édition des pictogrammes

Il est possible de retrouver en annexe [la page d'accueil de cette fonctionnalité](#), ainsi que [la page qui remercie l'utilisateur pour sa participation](#).

Page DicoPicto

Création d'un vocabulaire en pictogrammes ARASAAC

Rechercher un mot

bonjour

Valider

Effacer

Résultat pour le mot : "bonjour"



Figure 6 : Page de dictionnaire de pictogramme

Dans la page DicoPicto, réservée aux personnes disposant d'un compte permettant l'accès à la page. L'objectif de cette page est de récolter les retours des utilisateurs afin d'accumuler des données concernant quels pictogrammes sont les plus adaptés parmi les différentes propositions.

II.5 Impact environnemental et sociétal

Je vais présenter ici les impacts sur l'environnement et sur la société en ce qui concerne mon stage. Dans un premier temps, je vais analyser mon impact personnel sur l'environnement dans le cadre de mon stage.

En utilisant le [site de l'Agence De l'Environnement et de la Maîtrise de l'Énergie \(ADEME\)](#), j'ai pu estimer de l'impact environnemental du trajet qui sépare le bâtiment IMAG de mon domicile. Mon trajet couvrant une distance de 3,77 km, la quantité de CO₂ émise par mon trajet serait de 0,02 kg lorsque j'utilise le tramway, et de 0,07 kg lorsque j'utilise le bus. En temps normal, mon trajet est fait en tramway, cependant, en raison de travaux sur la ligne C de tramway de Grenoble, j'utilise le bus sur au moins le dernier quart de mon stage.

De plus, dans le cadre du stage, je travaille sur ma machine personnelle, qui nécessite d'être branchée sur secteur, pour un coût total de 200 watts / heures.

Il est important de noter que l'hébergement de la plateforme sur les serveurs du bâtiment IMAG représente aussi un coût énergétique, toutefois, des vérifications sont souvent effectuées sur les serveurs afin d'assurer qu'aucun hébergement inutile n'ait lieu, ce qui permet de ne pas gaspiller inutilement des ressources.

L'équipe GETALP, travaille aussi en collaboration avec l'Association Française du Syndrome de Rett (AFSR)

Le LIG collabore avec Alternative and Augmentative Communication for Everyone (AAC4ALL), qui traite de la Communication Alternative et Améliorée (CAA), elle regroupe tous les moyens techniques et stratégiques mis en place pour renforcer ou remplacer une communication orale ou écrite déficiente. Elle concerne un large panel de handicaps physiques ou cognitifs et se traduit en partie par une offre d'outils techniques très diversifiée.

Le LIG est aussi lié avec l'Union pour la gestion des établissements des caisses de l'Assurance Maladie (UGECAM).

III. RÉALISATIONS

La section Réalisations a pour objectif de détailler les tâches et les projets spécifiques auxquels j'ai contribué au cours de ma période de stage. Cette partie met en lumière les différentes étapes de mon travail, les défis rencontrés et les solutions apportées. Elle inclut une description précise des missions effectuées, des technologies et des outils que j'ai utilisés pour accomplir ces missions. De plus, cette section permet de démontrer comment mes réalisations ont contribué aux objectifs globaux de l'organisation et ont apporté une valeur ajoutée aux projets en cours.

III.1 Gestion de projet

Fonctionnement global au sein de l'équipe

Le travail s'effectuait en présentiel, dans un bureau dédié aux stagiaires au sein du Laboratoire d'Informatique de Grenoble, sur ma machine personnelle. Les horaires étaient flexibles, à condition de respecter un total de 35 heures par semaine. J'ai installé le projet et veillé à son bon fonctionnement sur ma machine.

Organisation du travail

Durant mon stage, les tâches à accomplir m'ont été communiquées par Mme. Macaire et M. Arrigo. La communication se faisait principalement par [discord](#)¹⁸. Des réunions hebdomadaires encadrées par M. Arrigo et communes à tous les stagiaires de l'équipe avaient lieu en fin de semaine, elles permettaient de définir les objectifs à suivre de chacun. De plus, des réunions ont eu lieu toutes les deux semaines (ou au besoin), avec Mme Macaire et M. Arrigo avaient lieu pour parler plus en détail de mon projet, ces dernières avaient lieu avec l'utilisation de la plateforme [Zoom](#)¹⁹.

Le travail en lui-même s'organisait en utilisant GitHub, j'ai fait un fork du projet d'origine, j'ai ensuite créé des branches pour chacune des fonctionnalités que j'ai réalisées, elles m'ont aussi servies à garder une sécurité lorsque j'effectuais des actions à risque pour le projet (notamment lors des changements de versions).

III.2 Développement des fonctionnalités.

Page SpeechToPicto

La première étape de mon stage était de créer la page dans laquelle se trouveront les fonctionnalités à implémenter au cours du stage. Pour ce faire, j'ai créé un component Angular dédié à cette page.

Un composant (du nom anglais plus couramment utilisé "Component") est une classe TypeScript*, cette structure présente divers avantages que je vais présenter.

Encapsulation de la Logique et de l'Interface Utilisateur :

- Les composants Angular encapsulent à la fois la logique et l'interface utilisateur d'une partie spécifique de l'application. Cela permet de découper l'application en modules distincts, rendant le code plus organisé et plus facile à maintenir.

Réutilisabilité :

- Un des avantages majeurs des composants est leur réutilisabilité. Une fois créé, un composant peut être utilisé dans différentes parties de l'application sans avoir à réécrire le code, ce qui améliore l'efficacité du développement et la cohérence de l'interface utilisateur.

Modularité :

- Angular favorise une architecture modulaire où chaque composant représente une unité fonctionnelle autonome. Cette modularité simplifie le développement, les tests et la maintenance de l'application, car chaque module peut être développé et testé indépendamment avant d'être intégré dans l'application globale.

Gestion de l'État :

- Chaque composant peut gérer son propre état interne et communiquer avec d'autres composants via des propriétés d'entrée (input) et de sortie (output). Cela permet une gestion fine et localisée des données et des interactions utilisateur.

Séparation dans le code :

- En séparant la logique de présentation de la logique métier, les composants Angular facilitent la gestion et la compréhension du code. Le template [HTML²⁰](#) d'un composant contient la structure et le style de l'interface utilisateur, tandis que le fichier TypeScript gère les données et les méthodes associées.

Testabilité :

- Les composants Angular sont conçus pour être facilement testables. Grâce à leur encapsulation, il est possible de créer des tests unitaires pour vérifier le comportement des composants de manière isolée, assurant ainsi la fiabilité et la qualité du code.

L'architecture du composant comprend un HTML, un [CSS²¹](#), ainsi que deux fichiers TypeScript, comme il est possible de le voir dans la figure en annexe "architecture d'un composant", c'est donc dans ce composant que les fonctionnalités implémentées se retrouveront.

Front-End de la page

Afin que les fonctionnalités soient utilisables, il est important de créer une interface graphique permettant leur usage, c'est pourquoi j'ai pu m'appuyer sur une maquette donnant une idée de ce à quoi doit ressembler la place, vous pouvez la voir ci-après.

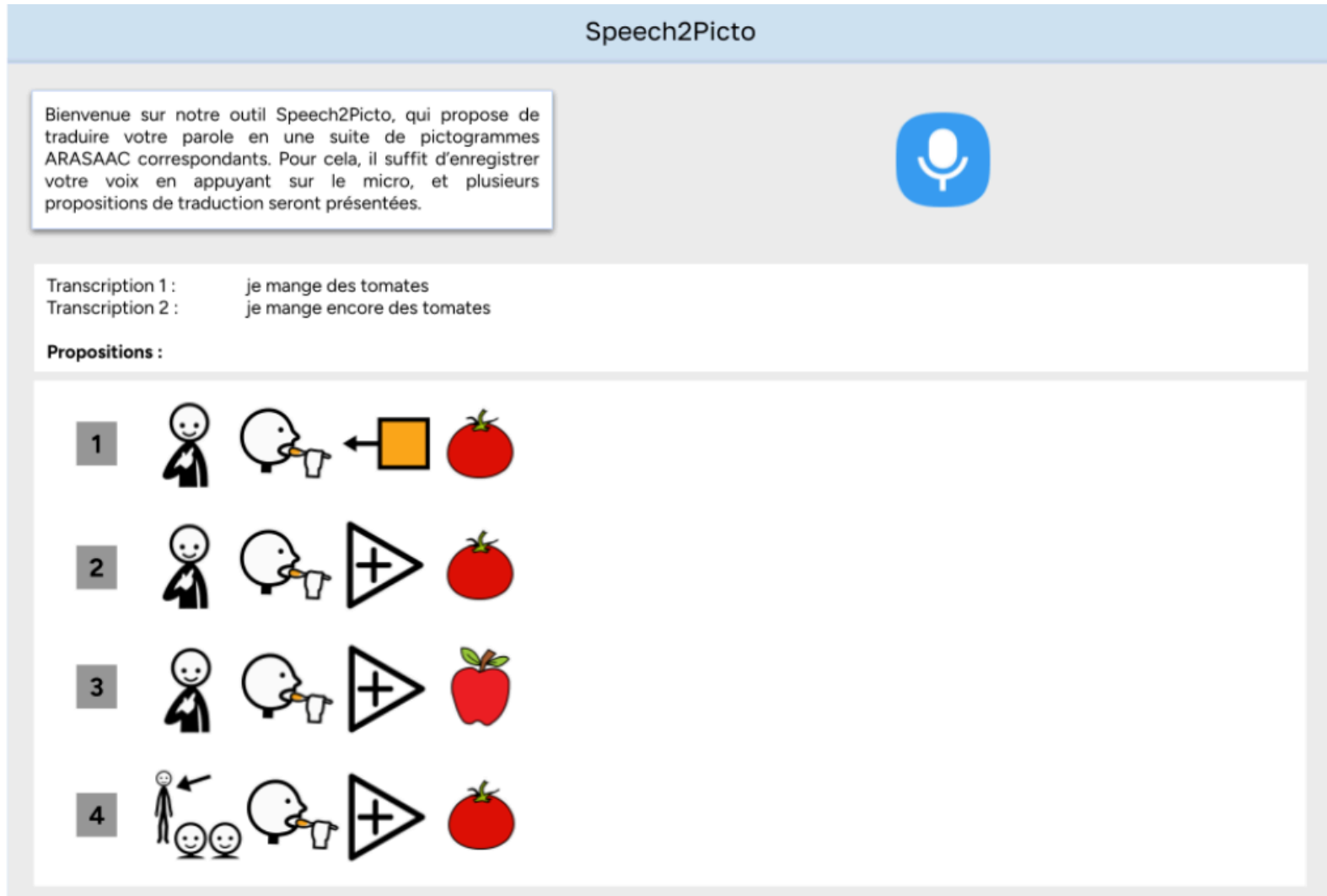


Figure 7 : Maquette de l'interface de la page SpeechToPicto

Je me suis donc basé sur cette maquette afin de produire une Interface Homme-Machine satisfaisant les besoins en lien avec l'implémentation des fonctionnalités. Vous pouvez voir ci-dessous la page.

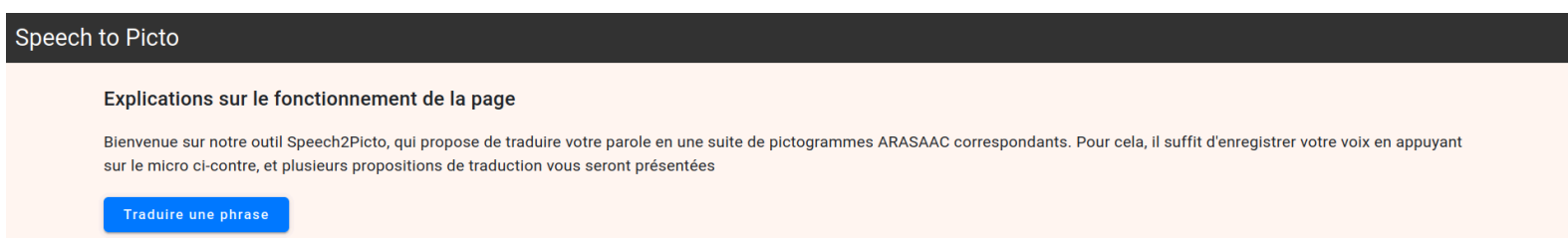


Figure 8 : Page SpeechToPicto

La page contient des instructions sur son fonctionnement. Le bouton “Traduire une phrase” permet d’accéder à la suite du programme.

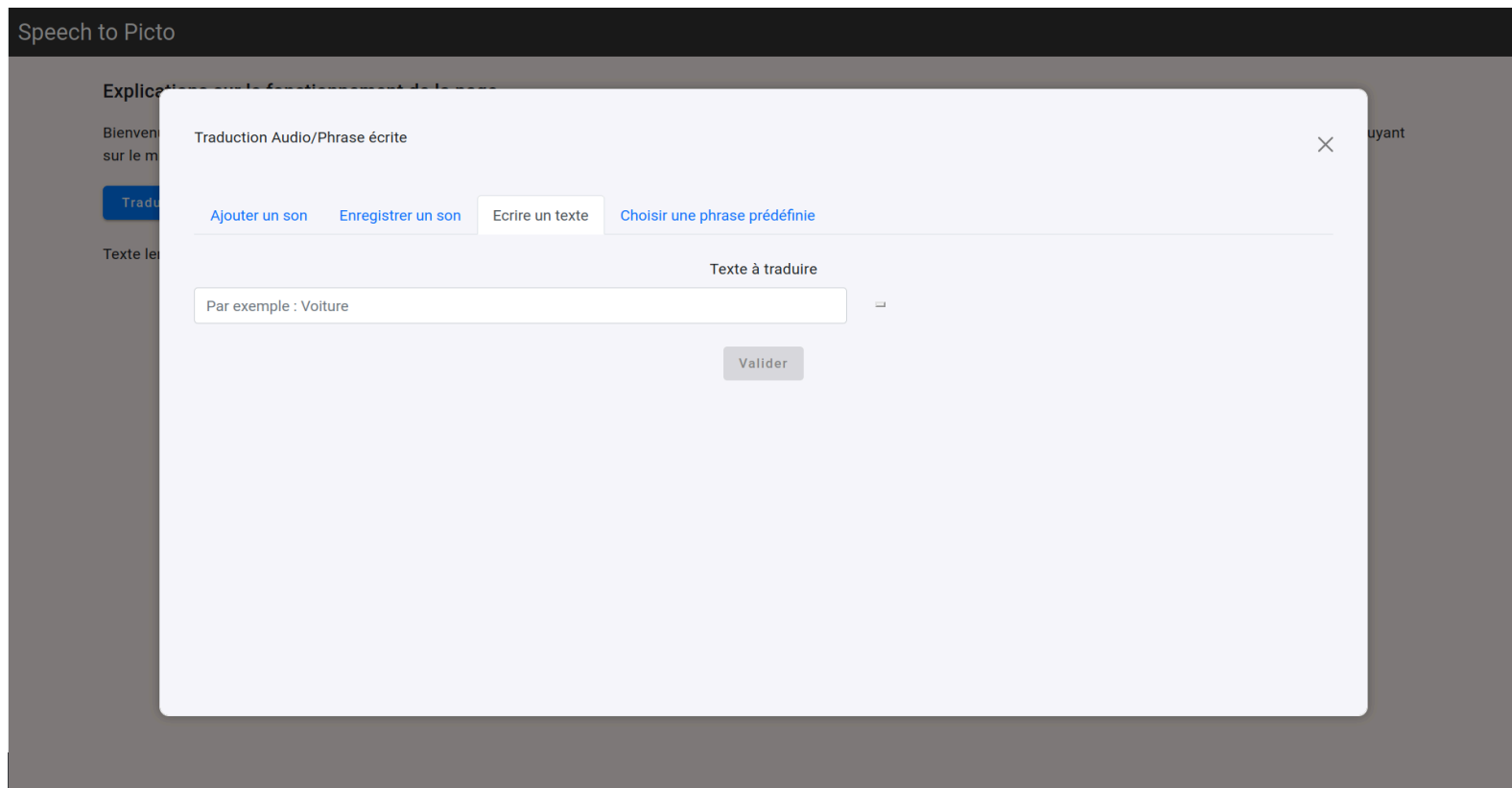


Figure 9 : Traduction à partir d'un texte

Lorsqu'on clique sur le bouton, un menu s'affiche, permettant de choisir la méthode selon laquelle l'utilisateur souhaite obtenir les pictogrammes. Dans l'image ci-dessus, il est possible de traduire depuis un texte écrit, comme dans la [page Picto](#) présentées précédemment. Un bouton permet également à l'utilisateur d'écouter la phrase écrite grâce au [service²²](#) MediaRecorder que je vais présenter dans la suite du rapport. Cependant, il est important de noter que cette fonctionnalité présente des problèmes en lien avec certains [Navigateurs Web²³](#) notamment Mozilla Firefox. Il est donc recommandé d'utiliser le navigateur Google Chrome, qui lui, ne présente aucuns problèmes lors de la lecture d'une phrase sous forme d'audio.

Dans un premier temps, j'ai appliqué à ce texte la lemmatisation vue préalablement, toutefois, il est prévu de passer par l'utilisation de différents scripts que je vais présenter.

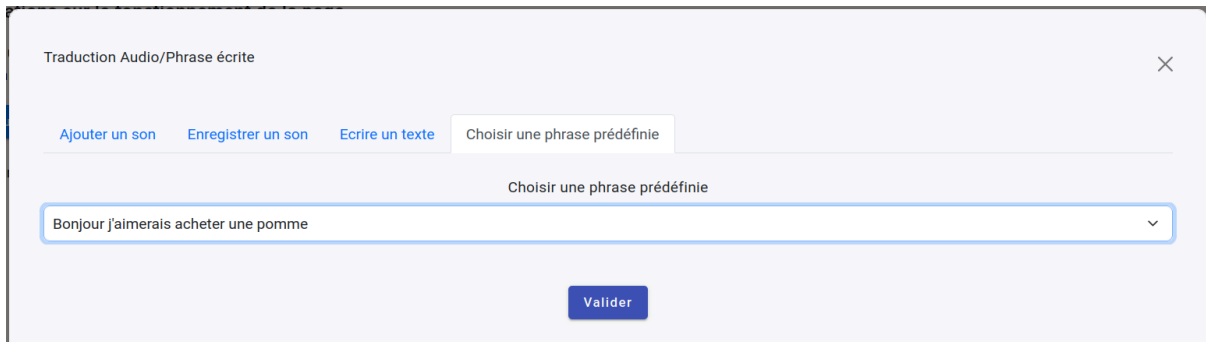


Figure 10 : Traduction à partir de phrases prédéfinies

L'utilisateur pourra aussi choisir des phrases prédéfinies (à décider à l'avenir) afin de les traduire en pictogrammes.

Enfin, je vais aborder la partie principale du stage, à savoir la Reconnaissance Automatique de la Parole.

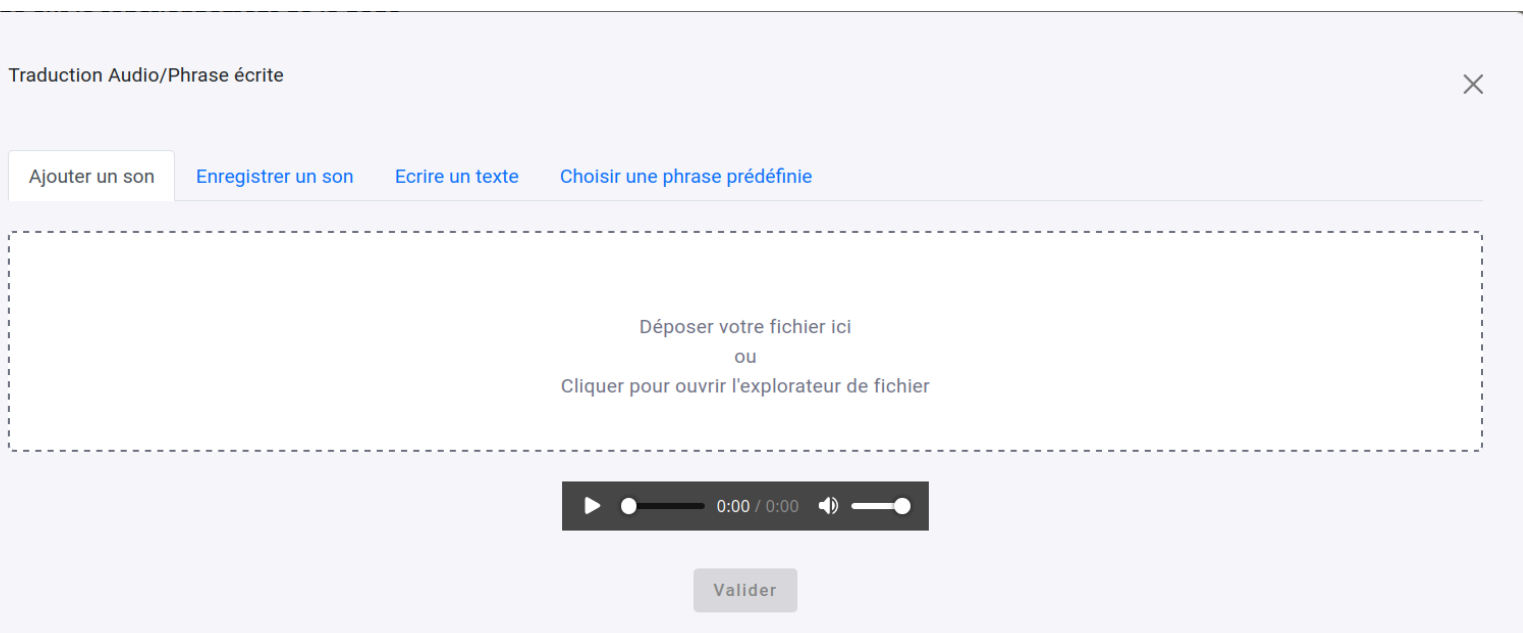


Figure 11 : Traduction à partir d'un fichier audio

Il est ici possible pour l'utilisateur de choisir un fichier audio, dans le but de le traduire sous forme de pictogrammes. L'utilisateur peut alors avoir un aperçu du fichier choisi depuis le lecteur prévu pour.

Traduction Audio/Phrase écrite



Ajouter un son

Enregistrer un son

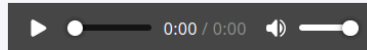
Ecrire un texte

Choisir une phrase prédéfinie

Pas d'enregistrement en cours

Nom de mon enregistrement

Par exemple : MonEnregistrement



Valider

Figure 12 : Traduction à partir de la voix

La dernière option possible, permet à l'utilisateur d'enregistrer sa propre voix grâce à son microphone afin d'en tirer un fichier audio qui sera utilisé pour la traduction en pictogrammes.

L'implémentation de ce menu m'a appris à partager des valeurs de variables entre plusieurs composants, chose que je n'avais pas faite auparavant. Pour ce faire, j'ai dû créer un service le permettant. Je vais donc expliquer ce qu'est un service ainsi que son utilité.

Dans le cadre du développement d'applications avec Angular, un service est une classe dédiée à l'exécution de tâches spécifiques qui ne sont pas directement liées à la gestion de l'interface utilisateur. Les services jouent un rôle crucial dans la structure et l'organisation des applications Angular, en facilitant la séparation des préoccupations et en promouvant la réutilisabilité du code.

- Séparation dans le code :

- Les services permettent de séparer la logique métier et les opérations complexes des composants. Cela signifie que les composants peuvent se concentrer sur la gestion de l'interface utilisateur, tandis que les services gèrent les opérations en arrière-plan telles que les appels API, la manipulation des données et d'autres tâches métiers.

- Réutilisabilité :

- Les services sont conçus pour être réutilisables à travers toute l'application. Une fois qu'un service est créé, il peut être injecté dans n'importe quel composant, directive ou autre service, réduisant ainsi la duplication de code et facilitant la maintenance.

- Gestion de l'État :

- Les services peuvent gérer l'état de l'application en conservant des données partagées entre différents composants. Par exemple, un service peut stocker les

informations d'un utilisateur connecté ou les données récupérées d'une API, permettant ainsi aux composants d'accéder et de réagir aux changements de cet état.

- **Communication entre Composants :**

- Les services facilitent la communication entre des composants qui ne sont pas directement liés par une relation parent-enfant. En utilisant des services, il est possible de partager des données et des événements entre différents composants de manière propre et efficace.

J'ai donc créé un service nommé `AudioTextFileShareService`, qui permet de faire la liaison entre les composants audio-text-file (le menu précédemment présenté) ainsi que le composant `speech-to-picto`. Ce service Angular, utilise `BehaviorSubject` de la bibliothèque [RxJS²⁴](#) pour gérer et partager des données textuelles lemmatisées entre différents composants de l'application. L'utilisation de `BehaviorSubject` permet de conserver la valeur courante du texte lemmatisé. La méthode `setLemmatizedText` permet de mettre à jour cette valeur partagée.

En implémentant ce service, j'ai acquis une meilleure compréhension des concepts suivants :

- Injection de dépendances dans Angular : En utilisant le décorateur `@Injectable`, j'ai appris à configurer et à fournir des services à l'échelle de l'application.
- Utilisation de RxJS pour la gestion des états : L'utilisation de `BehaviorSubject` m'a permis de comprendre comment gérer et diffuser des états partagés de manière réactive.
- Modularité et réutilisabilité du code : La création de services dédiés favorise la séparation des préoccupations et rend le code plus maintenable et réutilisable.

Pour faire simple, cela permet aux variables contenues dans le composant du menu d'être transmises au composant de la page, ce qui est essentiel pour pouvoir utiliser le fichier audio.

MediaRecorder

Les fonctionnalités en lien avec l'audio ont nécessité que j'implémente un service `MediaRecorder`, utilisant la librairie du même nom.

L'API `MediaRecorder` est une interface JavaScript fournie par les navigateurs web qui permet d'enregistrer des flux multimédia, comme l'audio ou la vidéo, directement à partir des périphériques de l'utilisateur, tels que le microphone et la caméra. Elle fait partie de l'API WebRTC (Web Real-Time Communication) et est particulièrement utile pour les applications web interactives qui nécessitent des capacités d'enregistrement.

L'intérêt d'utiliser `MediaRecorder` se retrouve dans son :

Accessibilité et Simplicité : `MediaRecorder` permet aux développeurs web d'intégrer des fonctionnalités d'enregistrement directement dans les applications web sans nécessiter de plugins tiers ou de logiciels supplémentaires. Cela simplifie le processus pour les utilisateurs finaux, qui n'ont pas à installer de logiciels supplémentaires.

Compatibilité Multi-plateforme : Étant une API standardisée, MediaRecorder fonctionne de manière cohérente sur différents navigateurs et systèmes d'exploitation, assurant une expérience utilisateur uniforme.

Applications Riches en Fonctionnalités : Avec MediaRecorder, les développeurs peuvent créer une gamme variée d'applications, telles que des enregistreurs vocaux, des applications de podcast, des outils d'apprentissage des langues et des fonctionnalités de visioconférence avec enregistrement.

Traitement en Temps Réel : Les données capturées peuvent être traitées en temps réel, ce qui permet des applications avancées comme la reconnaissance vocale, l'analyse audio et la manipulation audio/vidéo en direct.

À l'aide de la figure suivante, je vais expliquer de manière claire le fonctionnement de MediaRecorder.

```
1 usage  EnzoVigne
startRecording() : void {
  this.audioChunks = [];
  navigator.mediaDevices.getUserMedia({constraints: {audio: true}})
    .then(stream : MediaStream => {
      this.mediaRecorder = new MediaRecorder(stream);
      this.mediaRecorder.start();

      this.mediaRecorder.addEventListener(type: "dataavailable", listener: event : BlobEvent => {
        this.audioChunks.push(event.data);
      });

      this.mediaRecorder.addEventListener(type: "stop", listener: () : void => {
        this.audioBlob = new Blob(this.audioChunks, options: {type: "audio/wav"});
        this.audioUrl = URL.createObjectURL(this.audioBlob);
        this.audio = new Audio(this.audioUrl);
      });
    });
}
```

Figure 13 : Code MediaRecorder

Lorsqu'une application web utilise MediaRecorder, elle commence par demander l'autorisation d'accéder au microphone de l'utilisateur. Une fois cette autorisation obtenue, MediaRecorder est initialisé avec le flux audio provenant du microphone.

Pendant l'enregistrement, MediaRecorder capture des segments de données audio à intervalles réguliers. Ces segments, appelés "blobs", sont collectés et peuvent être manipulés en temps réel par l'application.

À la fin de l'enregistrement, MediaRecorder finalise les données capturées en générant un blob final. Ce blob peut être transformé en une URL utilisable, ce qui permet à l'application de proposer à l'utilisateur de lire ou de télécharger l'enregistrement directement depuis le navigateur.

En résumé, MediaRecorder simplifie l'intégration d'une fonctionnalité d'enregistrement audio dans les applications web, en offrant une solution native pour capturer, stocker puis traiter des données audio en temps réel, contribuant ainsi à enrichir les interactions utilisateur et les fonctionnalités des applications modernes.

Intégration des scripts Python

Différents scripts python sont implémentés ou en cours d'implémentation dans le projet, notamment Whisper, qui a été mentionné et expliqué précédemment dans le rapport ([voir Technologies utilisées - Python](#)).

J'ai donc dans un premier temps testé les différents scripts de Reconnaissance Automatique de la Parole. Cependant, bien que j'aie pu utiliser en local ces scripts, mes intégrations en ce qui concerne ces scripts ne sont pas compatibles avec la gestion de serveur qu'il est censé y avoir. En effet, les scripts, en particulier Whisper, peuvent être assez gourmands en ressource (particulièrement pour le [GPU²⁵](#)). Comme il est possible de le [voir en annexe](#), Whisper dispose de plusieurs versions plus ou moins lourde, plus la version est lourde, plus le résultat de sortie du script sera correct, en revanche les temps de réponses peuvent augmenter fortement.

Il est donc important que ces scripts soient exécutés sur le serveur ayant des GPU dédiés afin de garantir des temps de réponses corrects. Mes tuteurs et moi avons ainsi fait une demande pour avoir un accès aux serveurs ainsi qu'aux ressources nécessaires, toutefois, je n'ai pas encore pu y accéder dans le but d'y implémenter les scripts.

En cours

Au moment de la rédaction de ce rapport, je suis en pleine phase d'implémentation du [front-end²⁶](#) destiné à afficher les images correspondant aux requêtes des utilisateurs. Pour atteindre cet objectif, et comme je l'ai expliqué précédemment, j'ai opté pour l'utilisation de l'API Arasaac au lieu de me limiter à des images stockées localement. Cette approche me permet de bénéficier d'une banque d'images plus diversifiée et constamment mise à jour, ce qui améliore significativement la pertinence et la qualité des visuels proposés aux utilisateurs. En effet, l'API Arasaac offre un accès à une vaste collection d'images actualisées

régulièrement, ce qui garantit que les utilisateurs disposent toujours des ressources graphiques les plus pertinentes et récentes pour répondre à leurs besoins.

Afin de mieux comprendre ce qu'est concrètement une API, je vais détailler ici même ce que c'est concrètement. Une API (Application Programming Interface), ou interface de programmation d'applications en français, est un ensemble de règles et de protocoles qui permet à différentes applications logicielles de communiquer entre elles. Elle définit la manière dont les composants logiciels doivent interagir, facilitant ainsi l'intégration et la fonctionnalité entre divers systèmes.

Une API fonctionne comme un intermédiaire entre deux applications. Lorsqu'une application (le client) envoie une requête à une autre application (le serveur), l'API reçoit cette requête, la traite, et renvoie les données ou les actions demandées. Les requêtes et les réponses sont généralement échangées via des formats de données standardisés comme [JSON](#)²⁷ ou [XML](#)²⁸.

Les avantages de l'utilisation d'une API sont :

- **Modularité** : Permet de séparer les différentes fonctions d'une application en modules indépendants.
- **Réutilisabilité** : Favorise la réutilisation de fonctionnalités à travers différentes applications.
- **Interopérabilité** : Facilite la communication entre systèmes disparates.
- **Efficacité** : Réduit le temps de développement en utilisant des composants préexistants.

L'API Arasaac est un exemple d'API web qui permet aux développeurs d'accéder à une banque d'images pictographiques et de symboles visuels. En intégrant cette API dans une application, les développeurs peuvent permettre à leurs utilisateurs de rechercher, afficher et utiliser ces images de manière dynamique sans avoir à les stocker localement. Cela assure que les utilisateurs ont toujours accès aux images les plus récentes et pertinentes.

Travail restant

Il reste deux tâches à accomplir afin que le projet soit totalement terminé, à savoir :

- Implémenter dans le serveur les fonctionnalités en lien avec les scripts Python
- Implémenter les retours utilisateurs

En ce qui concerne les fonctionnalités en lien avec les scripts Python, comme mentionné plus tôt dans le rapport (voir [Réalisations - Intégration des scripts Python](#)), il reste à implémenter les scripts dans le serveur. Malheureusement, le serveur de la plateforme ne permet pas encore de communiquer avec le serveur GPU pour le moment (ce qui est nécessaire pour obtenir les résultats des scripts), il faut donc attendre que les modifications nécessaires soient apportées par l'équipe Moyens Informatique (MI) du LIG.

Ensuite, pour ce qui est des retours des utilisateurs, il s'agit de la finalité de la plateforme est non seulement de permettre aux personnes en situation en handicap langagier d'avoir un outil permettant la communication sous forme de pictogramme. Mais aussi de récolter des avis des

utilisateurs quant à la performance de l'emploi de la Reconnaissance Automatique de la Parole pour la retranscription sous forme de pictogrammes.

Il reste donc à créer une interface permettant d'enregistrer les différents retours que pourraient avoir les utilisateurs, avec un système de note par exemple.

Une dernière chose qui pourrait être très intéressante à implémenter, serait la réalisation de tests, notamment des tests unitaires.

Un test unitaire est une méthode de vérification qui consiste à tester individuellement chaque composant ou unité de code, souvent une fonction ou une méthode, pour s'assurer qu'elle fonctionne correctement. Ces tests sont généralement automatisés et conçus pour vérifier des aspects spécifiques et limités du comportement du code.

L'objectif de la réalisation de tests unitaires est de :

- **Détection des Erreurs** : Les tests unitaires permettent de détecter les erreurs dès les premières phases du développement, ce qui réduit les coûts de correction.
- **Facilité de Maintenance** : En vérifiant que chaque unité de code fonctionne comme prévu, les tests unitaires facilitent la maintenance et les mises à jour du code. Ils permettent de s'assurer que les modifications n'introduisent pas de nouvelles erreurs.
- **Documentation Vivante** : Les tests unitaires servent de documentation vivante pour le code. Ils montrent comment chaque composant est censé fonctionner, ce qui aide les nouveaux développeurs à comprendre le projet plus rapidement.
- **Confiance dans le Code** : Avoir une suite de tests unitaires robuste augmente la confiance des développeurs dans le code, car ils savent que les fonctionnalités critiques sont vérifiées en continu.

CONCLUSION

Au cours de mon stage, j'ai eu l'opportunité de travailler sur le développement d'un outil utilisant la Reconnaissance Automatique de la Parole.

Ce projet m'a permis de développer de nombreuses compétences essentielles pour mon parcours académique et professionnel. Cette expérience m'a également permis d'améliorer ma capacité à travailler de manière autonome tout en restant rigoureux et méthodique.

J'ai pu renforcer mes connaissances dans des outils que je connaissais déjà, notamment Angular. J'ai donc pu appliquer ce que j'ai appris lors de mon apprentissage au sein du BUT informatique de l'IUT2 de Grenoble. De plus, j'ai aussi pu aborder des choses avec lesquelles je n'étais pas à l'aise auparavant, particulièrement la communication entre différents langages (en l'occurrence les scripts JS ou Python avec Angular qui utilise TypeScript).

Ce stage a été une expérience agréable grâce à son ambiance, son sujet intéressant, une équipe accueillante. J'ai pu explorer de nouvelles technologies et renforcer mes connaissances actuelles. J'aurais préféré pouvoir venir à bout du projet, toutefois, je suis satisfait de savoir que mon travail aura une utilité réelle.

GLOSSAIRE

Angular

Framework de développement web front-end basé sur TypeScript, maintenu par Google, utilisé pour créer des applications web.

API (Application Programming Interface)

Interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

Composant

En développement web, un composant est une unité modulaire réutilisable qui encapsule une partie de l'interface utilisateur.

CSS (Cascading Style Sheets)

Langage utilisé pour décrire la présentation d'un document écrit en HTML ou XML.

Discord

Plateforme de communication qui offre des services permettant la création de serveurs, groupes, afin de communiquer à l'oral ou à l'écrit.

Framework

En programmation informatique, un framework est un ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel, c'est-à-dire une architecture.

Front-End

Le développement web frontal correspond aux productions HTML, CSS et JavaScript d'une page internet ou d'une application qu'un utilisateur peut voir et avec lesquelles il peut interagir directement.

GPU (Graphics Processing Unit)

Un processeur graphique (GPU) est un circuit électronique capable d'effectuer des calculs mathématiques à grande vitesse.

Handicap langagier

Déficience qui affecte la capacité d'une personne à comprendre ou à utiliser le langage parlé ou écrit.

HTML (HyperText Markup Language)

Langage de programmation standard utilisé pour créer des pages web.

IDE (Integrated Development Environment)

Environnement de développement intégré, outil qui fournit des fonctionnalités de développement de logiciels (ici IntelliJ Ultimate).

Intelligence Artificielle (IA)

Champ de l'informatique qui se concentre sur la création de systèmes capables de réaliser des tâches nécessitant normalement l'intelligence humaine, comme la reconnaissance d'image, la compréhension du langage naturel et la prise de décision.

JavaScript

Langage de programmation utilisé principalement pour le développement web, permettant de rendre les pages web interactives.

JSON (JavaScript Object Notation)

Format léger d'échange de données, facile à lire et à écrire pour les humains, et facile à analyser et à générer pour les machines.

Lemmatisation

Processus de réduction des mots à leur forme de base ou canonique ("mangais" deviendrait alors "manger").

Navigateurs Web

Logiciels utilisés pour accéder et visualiser des pages web

Pictogrammes

Symboles graphiques utilisés pour représenter des mots ou des concepts.

Python

Langage de programmation interprété, polyvalent, largement utilisé en intelligence artificielle, science des données, développement web.

Reconnaissance Automatique de la Parole (RAP)

Technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

RxJS (Reactive Extensions for JavaScript)

Bibliothèque pour la programmation réactive utilisant des Observables, facilitant la composition de programmes asynchrones ou basés sur des événements.

Service

En développement Angular, un service est une classe qui a pour but de s'occuper de la logique métier, et qui n'interagit pas avec l'interface.

Single Page Application (SPA)

Application web qui charge une seule page HTML et met à jour dynamiquement le contenu en fonction des interactions de l'utilisateur.

Traitement Automatique des Langues (TAL)

Autre terme pour le traitement du langage naturel.

Traitement du Langage Naturel (NLP)

Sous-domaine de l'Intelligence Artificielle qui se concentre sur l'interaction entre les ordinateurs et les humains en utilisant le langage naturel.

TypeScript

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript.

URL (Uniform Resource Locator)

Adresse web utilisée pour accéder à des ressources sur Internet.

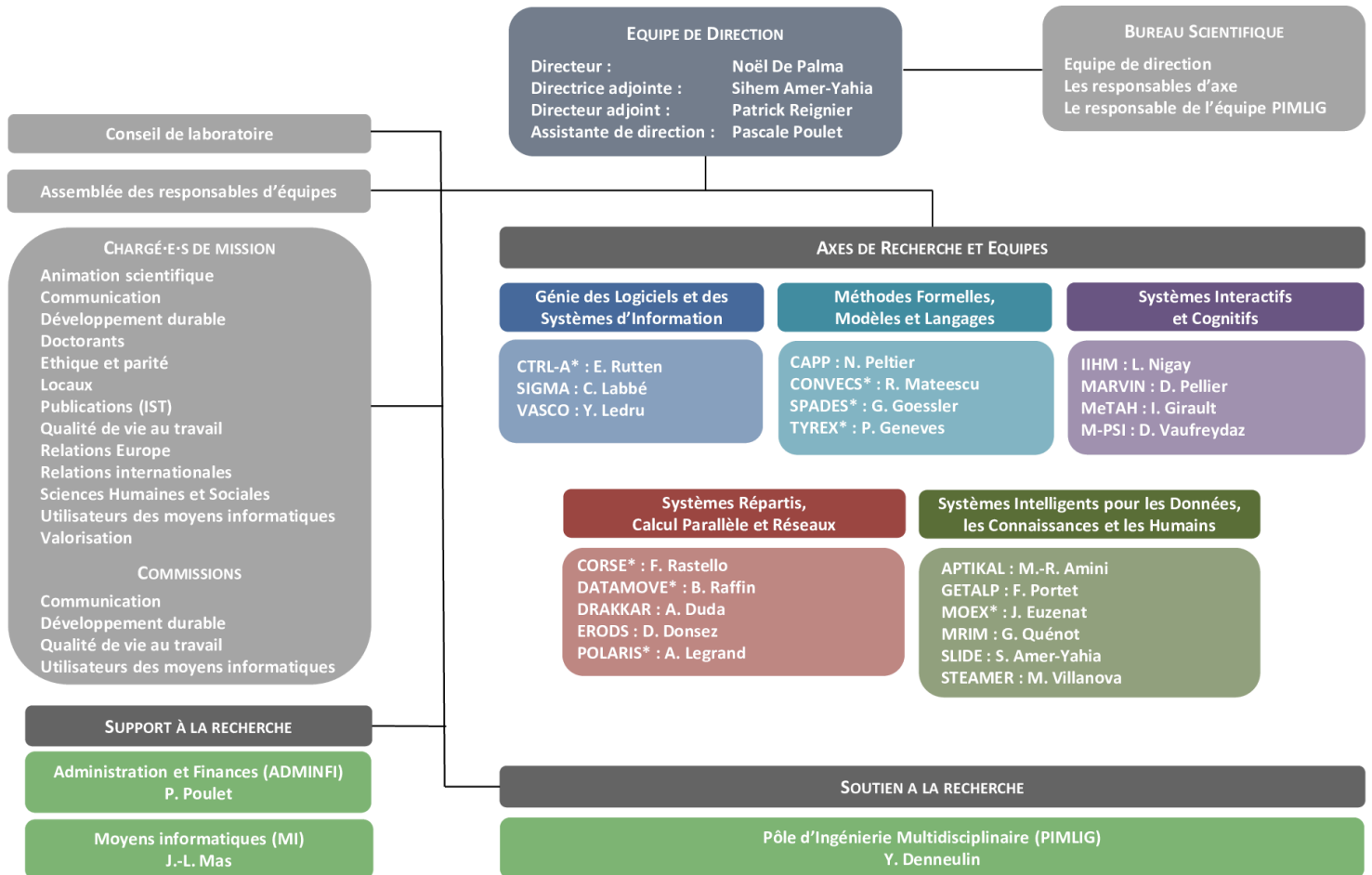
XML (Extensible Markup Language)

Langage de balisage extensible utilisé pour structurer des données, souvent utilisé dans les échanges de données entre applications.

Zoom

Plateforme de visioconférence populaire.

ANNEXES



* Equipe projet commune INRIA

UMR5217 – www.liglab.fr – septembre 2023

Annexe 1 : Organigramme du LIG

Post-édition de phrases traduites en pictogrammes ARASAAC

Présentation de la tâche

La tâche que nous vous présentons est une tâche de post-édition, qui consiste à vérifier les traductions proposées par un outil.

À l'écran, vous serez proposés des phrases suivies de leur traduction en pictogrammes ARASAAC. Votre but est simplement de les corriger. Les instructions sont données à la page suivante.

Ce travail est intégré au projet de thèse de Cécile MACAIRE, intitulé "Traduction automatique de la parole vers des pictogrammes". Le but est de transcrire la parole en une séquence de pictogrammes. Cette thèse est financée par le projet PROPICTO, décrit ci-dessous.

Projet PROPICTO

PROPICTO est un projet bilatéral franco-suisse, financé respectivement par l'Agence Nationale française pour la Recherche (ANR) et le Fonds National Suisse de la recherche scientifique (FNS). Ce programme de recherche de 4 ans est mené en collaboration entre le Département de Technologie de la Traduction (TIM) de l'Université de Genève et le Groupe d'Étude de la Traduction Automatique et du Traitement Automatique des Langues et de la Parole (GETALP), rattaché au Laboratoire d'Informatique de Grenoble (LIG). L'objectif global de PROPICTO (PROjection du langage Oral vers des unités PICTOgraphiques) est de créer des systèmes de traduction permettant de convertir de la parole en une séquence de pictogrammes. Ces systèmes permettront ainsi d'améliorer l'accès à la communication pour les non-francophones (allophones) ou les personnes en situation de handicap langagier pour communiquer avec leur proche, et le personnel médical.

Aucune donnée personnelle n'est récupérée. Pour toutes questions relatives au projet, vous pouvez nous contacter à l'adresse suivante : corpuspropicto@univ-grenoble-alpes.fr

☐ J'ai lu et j'accepte les termes et conditions

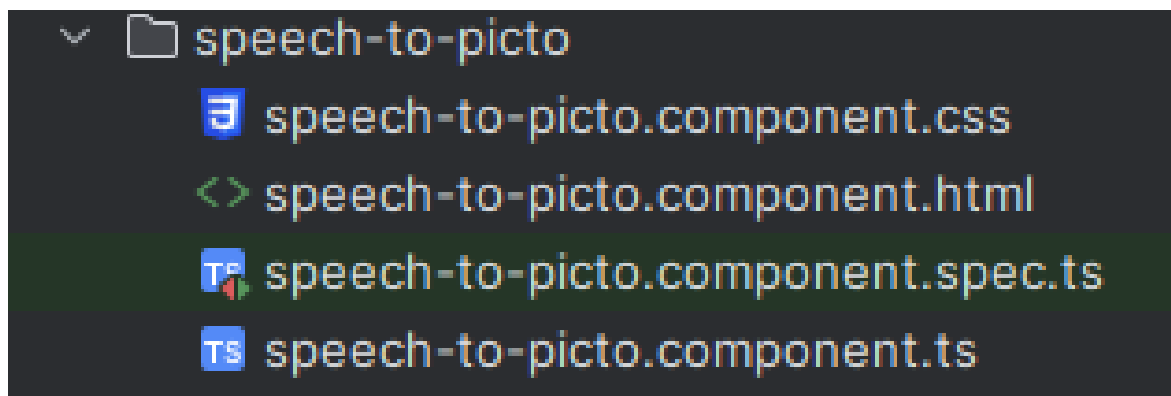
Commencer

*Annexe 2 : Page de Post-Edition - Accueil*

Post-édition de phrases traduites en pictogrammes ARASAAC

Merci de votre participation !

Pour toutes questions relatives au projet, vous pouvez nous contacter à l'adresse suivante : corpuspropicto@univ-grenoble-alpes.fr

*Annexe 3 : Page de Post-Édition - Remerciement*

Annexe 4 : Architecture d'un composant

Size	Parameters	English-only	Multilingual
tiny	39 M	✓	✓
base	74 M	✓	✓
small	244 M	✓	✓
medium	769 M	✓	✓
large	1550 M	x	✓
large-v2	1550 M	x	✓
large-v3	1550 M	x	✓

Annexe 5 : Poids de Whisper

RÉFÉRENCES BIBLIOGRAPHIQUE ET SITOGRAPHIQUE

[1] LigLab

<https://www.liglab.fr/fr/presentation/organigramme>

[2] Propicto

<https://propicto.unige.ch/>

[3] Lig-InterAActionPicto

<https://lig-interaactionpicto.imag.fr/#/picto>

[4] Site de l'ADEME

<https://agirpourlatransition.ademe.fr/particuliers/bureau/deplacements/calculer-emissions-carbone-trajets>

Résumé court :

Le stage s'est déroulé au sein de l'équipe nommée Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole (GETALP) au sein du Laboratoire d'Informatique de Grenoble (LIG). Le but de ma mission était d'implémenter dans une plateforme déjà existante des fonctionnalités de Reconnaissance Automatique de la Parole (RAP), qui vise à pouvoir retranscrire la parole sous forme de texte, cela est fait à l'aide de différents scripts notamment Whisper, basé sur l'Intelligence Artificielle (IA). L'objectif étant de proposer une plateforme permettant aux personnes en situation de handicap langagier de communiquer par le biais de pictogrammes trouvables grâce à l'API Arasaac. Le projet en lui-même s'appelle InterAActionPicto et est développé en Angular, un framework basé sur TypeScript. Mon travail a donc consisté à implémenter ces fonctionnalités, dans un premier temps en réalisant une interface graphique permettant de les exploiter, et dans un second temps à implémenter les scripts permettant le traitement de l'écrit ou de la parole afin de générer les pictogrammes souhaités.

Mots Clés :

GETALP, LIG, RAP, Whisper, IA, handicap langagier, pictogrammes, API, Arasaac, InterAActionPicto, Angular, framework, TypeScript.

Abstract :

Development of automatic speech recognition functions

Enzo Vigne.

abstract: As part of the Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole (GETALP) team at the Laboratoire d'Informatique de Grenoble (LIG). My mission was to integrate automatic speech recognition (ASR) functionalities into an existing platform in order to help people with speech disorders by transcribing speech into text using AI tools such as Whisper. The project, called InterAActionPicto, uses the Angular framework based on TypeScript. The implementation involved developing a graphical interface to enable users to access ASR functionality and integrating Python scripts, including Whisper, to process speech and text, generating pictograms via the Arasaac API. Firstly, I managed to develop an interface that allows users to use the aforementioned functionalities, and then to implement scripts that process audio files, although it is also possible to use the written word. All this, with the aim of generating the series of pictograms corresponding to the user's request.

Keywords :

GETALP, LIG, ASR, AI, Whisper, InterAActionPicto, Angular, framework, Python, scripts, pictograms, Arasaac, API.