Rapport de la 1ère partie du Projet Long : Interaction Humain-Drone

Soumis par les étudiants :
Saloua Naama
Théo Tournier
Wilfried L. Bounsi
Joêl Roman KY
Rachid Elmontassir

Encadrés par :
Coline Moinet
Sandy Vaslon

Année académique 2020/2021

Table des matières

Chapitre 1

Introduction

Les drones sont aujourd'hui utilisés dans de nombreux domaines tels que l'agriculture, la détection de feu de forêt, la surveillance, etc. L'idée de l'entreprise SII est de développer un drone qui pourrait aider les personnes en détresse (perdues ou blessées). Le projet consiste donc à programmer un drone pour être capable de détecter des personnes qui ont besoin d'aide et de converser avec celles ci. La caméra embarquée du drone servira pour la détection des personnes. Comme ce projet débute au sein de SII, tous les choix seront pris de manière à simplifier le plus possible la réalisation technique. L'objectif étant avant tout de réaliser un prototype qui fonctionne au détriment d'hypothèses non réalistes.
Le drone se déplacera donc dans un milieu montagneux à la recherche d'une personne à secourir, tel qu'une personne debout mais blessée. La partie navigation ne sera pas traitée dans ce projet car elle est déjà développée dans une équipe de SII. Nous considérerons donc que le drone se déplace de manière autonome.
Le projet va être traité en 2 parties, tout d'abord la partie identification et rapprochement du drone vers la personne en détresse, puis la partie communication avec la victime avec le développement d'un chatbot.

Chapitre 2

État de l'art

2.1 Positionnement face à une personne

A l'heure actuelle, il n'existe pas de traces dans la littérature d'un drone secouriste qui vient se positionner face à une personne pour lui venir en aide. Il existe cependant de multiples articles traitant des sujets de la reconnaissance des personnes par drone, et même particulièrement de personnes en détresse, tout en sachant que l'entreprise SII à déjà développé sa solution de détection que nous expliciterons plus tard. Pour le positionnement de la personne il existe par contre moins de papiers de recherches. A partir de la lecture de certains papiers nous avons eu 2 idées pour détecter les personnes en détresse : le survol à haute altitude et le survol à basse altitude.

2.1.1 Survol à hautes altitudes

Survoler à hautes altitudes (plusieurs dizaines de mètres) la zone et appliquer une série de filtres colorimétries pour détecter des personnes ([1]). Ces filtres ont pour but de convertir l'image initialement dans l'espace RGB dans l'espace YCbCr, qui est réputé pour être utilisé dans la reconnaissance des hommes. Une fois le filtre appliqué il faut définir un seuil qui est propre au milieu de recherche et à la caméra (mais les auteurs proposent des valeurs pour leur caméra) qui permet de définir les points à conserver et qui seraient potentiellement des victimes.
L'article [2] développe une méthode à base d'une Raspberry Pi3 équipée d'une Caméra V2 d'une résolution de 8MP. Cette méthode détecte une personne à partir d'un drone en vol à 12 m et le positionne juste à côté dans le cadre des livraisons par drones.
L'article [3], utilise des réseaux de neurones convolutifs pour détecter des personnes dans des zones de terrain à structures particulières, couplées à des techniques de filtrage afin de sélectionner les zones pouvant contenir un humain.
Suite à la détection de la personne en détresse il faut que le drone amorce une phase de rapprochement avec la victime. L'objectif est de placer le drone à une distance relativement proche (qui devra être définie durant le projet) pour que le drone et la personne puissent échanger convenablement. Les solutions imaginées sont :

1. Descendre par palier, en visant un point proche de la personne détectée. Cette méthode est inspirée des articles [1] et [4] qui demande de calculer l'angle entre la verticale du drone et la victime sur l'image (voir 2.2). A partir de l'angle on va choisir un point proche de la victime et placer le drone au dessus, et descendre de quelques mètres. Puis on réitère l'opération car on sait que le drone peut légèrement dévier de la verticale lorsqu'il descend à cause du vent par exemple.

2. Descendre par palier mais cette fois ci en centrant la personne dans l'image. On réitère le processus jusqu'à une certaine distance (assez faible) au dessus de la victime. Puis on effectue un dernier déplacement pour se placer à une faible distance en face de la victime.

Dans la littérature, il existe peu d'articles traitant d'une descente pas-à-pas d'un drone en se basant sur les images acquises par la caméra. Notre idée serait d'utiliser les techniques d'atterrissage utilisant les images acquises par la caméra du drone, mais en stabilisant le drone à une certaine hauteur au dessus du sol. L'article [4], utilise un algorithme qui détecte une zone d'atterrissage (marqueur) et se déplace de sorte à pouvoir atterrir de façon optimale sur le marqueur. L'article [2], essaie de proposer une solution permettant de détecter une personne d'un point de vue aérien et effectuer un atterrissage à côté de celle-ci en vue d'effectuer une livraison. Cette solution s'affranchit du besoin d'un marqueur et utilise les coordonnées GPS de la position de la personne afin d'amorcer une descente à une certaine distance

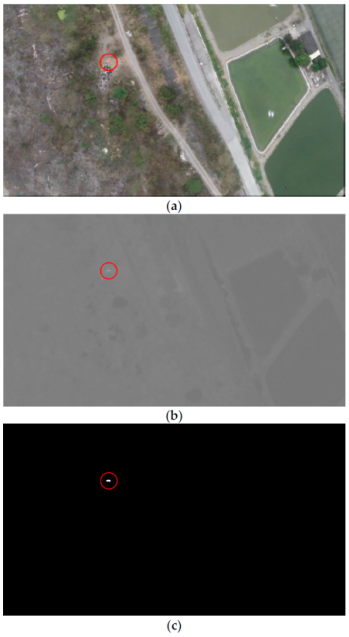

FIGURE 2.1 – **(a)** Image prise par le drone en RGB **(b)** Image dans l'espace YCrCb **(c)** Image binaire avec le seuil

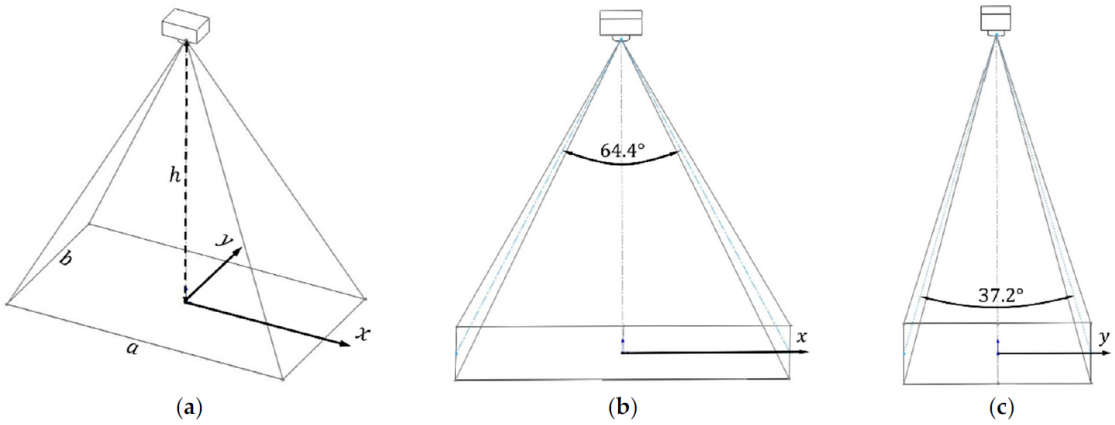

FIGURE 2.2 – Représentation dans l'espace de l'angle de vue de la caméra (les angles ne sont pas ceux de la caméra du Bebop2)

de la personne détectée. Cette solution repose sur des versions allégées et plus facilement embarquables, d'algorithme de détection comme *YOLO*, SSD et de traitement d'images (OpenCV).

2.1.2 Survol à basses altitudes

L'entreprise SII à déjà un produit qui détecte déjà les personnes et qui consiste à voler à basse altitude avec une vue légèrement plongeante. Cette détection se base sur la librairie *PoseNet* ([5]) et analyse les images en flux continu.
A partir de la détection de la personne réalisée via *PoseNet*, il faudra ensuite donner les bonnes consignes au drone pour se déplacer dans la bonne direction. Pour ce faire nous avons imaginé une technique qui se basera sur les boites tracées par le réseau de neurones utilisé pour la détection. L'idée est de situer le centre de cette boite (moyenne sur x et y des coordonnées des 4 coins de la boite) dans l'image : donc dire si il se situe à droite, à gauche ou au milieu. Évidement on se doute que le milieu ne sera presque jamais atteint, donc on diviserait l'image en 3 parties (pas forcément égales) pour permettre au drone de situer la victime et en déduire s'il doit tourner, et si oui de quel côté.

2.1.3 Estimation de la distance drone-victime

Dans la littérature, les articles traitant de l'estimation de la distance sont basés sur des détections d'objets via des boxes et estiment la distance à partir de ceux-ci. La méthode DisNet [6], par exemple utilise une caméra pour effectuer l'estimation de la distance ainsi qu'une caméra mono-oculaire, contrairement aux nombreuses méthodes de la littérature basée sur deux caméras stéréos.
La méthode DisNet est basée sur un réseau de neurones plutôt simple car il ne possède que 3 couches de 100 neurones chacune, le principe de ce réseau consiste à déterminer une relation entre la taille des boites identifiées par un réseau tel que *YOLO* ou *PoseNet*, tout en connaissant la taille de référence d'une boite pour un homme.

2.2 Chatbot pour recueillir ou donner des informations

2.2.1 Transcription

Malgré les grandes avancées dans le domaine, la reconnaissance de la parole reste un défi technique important [7]. Cependant cette dernière est désormais suffisamment bien maîtrisée pour donner des résultats satisfaisants suivant le type d'usage (dialogue de commande ou dictée vocale). LINAGORA développe un moteur de reconnaissance automatique de la parole (ASR, Automatic Speech Recognition) nommé LinSTT. Ce dernier est basé sur la boîte à outils open-source Kaldi [8]. La figure 2.3 illustre la chaîne de traitement de LinTO [9].
Dans la partie communication nous allons plutôt nous concentrer sur le positionnement, le modèle Rasa et le *Questions-Answering*, par conséquence, nous allons utiliser le transcripteur de LinTO qui donne des résultats très intéressants selon LINAGORA.

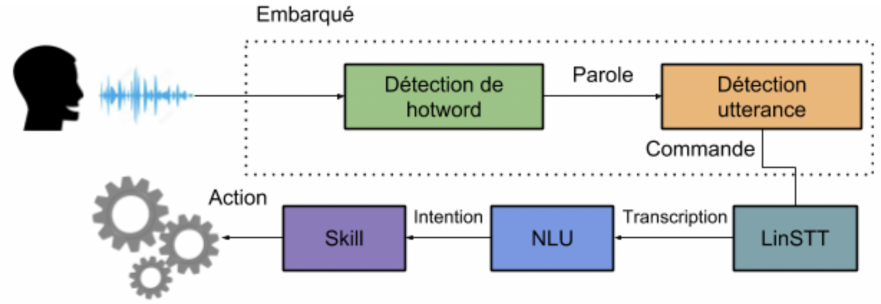

FIGURE 2.3 – Chaîne de traitement LinTO en mode commande.

2.2.2 Modèle Rasa

Les solutions permettant de développer un chatbot sont nombreuses, notamment celles des grandes entreprises telles que Amazon, IBM ou Google, ainsi que celles issues du monde open source. Rasa est l'un des meilleures frameworks de ce dernier, Il s'agit d'une entreprise allemande qui a rendu ses deux frameworks de machine learning Rasa NLU et Rasa Core open source.

- **Rasa NLU** : Une bibliothèque NLU(natural language understanding/compréhension du langage naturel) qui prend l'entrée de l'utilisateur, essaie de déduire l'intention et extrait les entités.
- **Rasa Core** : Responsable de la gestion de dialogue, il prend la structure de l'entrée du NLU et essaie de former un modèle de probabilité pour décider l'ensemble des actions à effectuer en se basant sur les entrées précédentes de l'utilisateur.

Parmi les concepts importants sur lesquels la solution Rasa se base il y a :

- **Intent** : (intention) l'objectif ou le but de l'entrée de l'utilisateur. Par exemple si l'entrée c'est : Réserver une table demain soir au restaurant X, L'intention sera : réserver une table
- **Entity** : (Entité) les informations utiles de l'entrée utilisateur qui peuvent être extraites. Sur l'exemple précédent, les entités peuvent être : Endroit : Restaurant X, Temps : Demain soir.
- **Actions** : Une opération effectuée par le bot, soit une simple réponse comme bonjour, comment allez vous ? ou bien une réponse personnalisée.
- **Stories** : C'est un échantillon des intéractions Utilisateur-Bot, définit à partir des intentions capturées et actions exécutées.
- **Domain** : qui englobe toutes les intentions ainsi que les réponses en hard code.

2.2.3 Modèle de langage pour les Question Réponses

Le drone secouriste que nous souhaitons développer doit être capable non seulement de prendre l'initiative de questionner une personne en détresse, mais aussi, de répondre au mieux à une question posée par la personne.

Ce besoin ouvre alors deux problématiques :
— Comment comprendre une question simple ou complexe ?
— Comment y apporter une réponse à la fois correcte et formulée naturellement ?

La tâche de répondre aux questions (Question Answering ou QA) est connue pour être une tâche difficile en traitement du langage naturel [10]. Les éléments clés du QA nécessite la capacité de comprendre la question et le contexte dans lequel la question est générée. La QA a été jugé difficile en raison de la nature dynamique du langage [10]. Cela a abouti à l'application de méthodes basées sur les données dans la réponse aux questions. Comme souvent en Machine Learning, l'idée est d'extraire la logique directement des données, plutôt que tenter de la coder dans les méthodes. Ceci paraît particulièrement pertinent dans ce contexte, car d'une part il est très difficile de formaliser le langage et d'autre part nous disposons d'énorme quantité de textes [11]. L'approche basée sur des règles était l'une des plus utilisées initialement des méthodes de premier plan pour les systèmes d'assurance qualité. Ces systèmes utilisaient les règles élaborées à partir de la sémantique grammaticale pour déterminer les réponses correctes pour une question donnée. Ces règles sont généralement fabriquées à la main et à l'aide d'heuristiques reposant sur le champ lexical et la sémantique du contexte [12].
Ces règles exploitent des modèles prédéfinis qui classent les questions en fonction du type de réponse. Ces règles grammaticales représentent le contexte sous forme d'arbres de décision et cela a été utilisé pour trouver le chemin qui mène à la bonne réponse [13]. Un inconvénient majeur des systèmes de réponse aux questions basés sur des règles était que les règles heuristiques devaient être manuellement conçues. Pour concevoir ces règles, une connaissance approfondie de la sémantique d'un langage était une nécessité [14]. Avec la croissance rapide des textes disponibles en ligne, l'importance des approches statistiques pour la QA a également augmenté. Ces approches reposent sur la prédiction de réponses basées sur des données. Comme ces méthodes sont capables de traiter l'hétérogénéité des données et exemptes de langages de requête structurés, elles ont été adaptées aux différentes étapes de l'assurance qualité [15].

Aujourd'hui, les modèles de langue neuronaux contextuels dits *Transformers* sont omniprésents en traitement automatique du langage, ils fournissent les meilleures perfomances dans toutes les différentes tâches, incluant le QA. Jusqu'à récemment, la plupart des modèles disponibles ont été entraînés soit sur des données en anglais, soit sur la concaténation de données dans plusieurs langues. L'utilisation pratique de ces modèles — dans toutes les langues sauf l'anglais — était donc limitée. La sortie récente de plusieurs modèles monolingues fondés sur BERT [16], notamment pour le français, a démontré l'intérêt de ces modèles en améliorant l'état de l'art pour toutes les tâches évaluées. Aujourd'hui, avec l'apparition de modèles tels que FlauBERT et CamemBERT [17] il existe maintenant de tels *Transformers* entièrement entrainés sur des données en français. Par ailleurs, avec le modèle CamemBERT, il a été démontré que l'utilisation de données à haute variabilité est préférable à des données plus uniformes. De façon plus surprenante, que l'utilisation d'un ensemble relativement petit de données issues du web (4Go) donne des résultats aussi bons que ceux obtenus à partir d'ensembles de données plus grands de deux ordres de grandeurs.

2.2.4 Sur-génération de données d'entraînement du modèle de langage

Dans le domaine des Chatbots, un enjeu crucial est d'avoir accès à des jeux de données suffisamment vastes et pertinents pour obtenir de bons résultats pour le modèle NLU. Gupta et al [18] travaillent sur la génération de paraphrases dans le cadre de la création d'une base de données d'entraînement de Chatbots. Ils pensent en effet que ce domaine rencontre des problèmes dus à un manque de données, à la presque obligation de passer par une étape manuelle de création de corpus et à une perte de temps conséquente dans la mise en place de cette création de données. Ils estiment que l'ère du Chatbot est arrivée et qu'il faut être en mesure de donner des outils efficaces dans le cadre de leur élaboration. Ils affirment que trois cas de figure sont possibles pour collecter ces précieuses données :
— L'ajout d'interactions réelles (*Crowd-Sourcing*, forums) ;
— L'ajout manuel d'exemples par le développeur ;
— La génération automatique de paraphrases sémantiquement équivalentes ;

La première méthode consiste à utiliser un matériel linguistique préexistant afin de l'incorporer dans la base de données. Le *Crowd-Sourcing* ainsi que l'acquisition de données depuis des forums en ligne en sont des exemples. Très peu de transformations, voire aucune, sont effectuées sur les données.
La troisième approche vise à générer, depuis un jeu de données restreint, des paraphrases (des phrases ou des expressions qui véhiculent le même sens en utilisant des formulations différentes) afin d'ajouter de

la diversité linguistique et d'améliorer le modèle.

Chapitre 3

Analyse critique des idées

3.1 Positionnement face à une personne

Les solutions ci-dessus sont très différentes, elles présentent toutes les deux des avantages et des inconvénients que nous allons présenter.

3.1.1 Survol à hautes altitudes

Le survol à haute altitude permet de surveiller une très grande zone et donc de trouver une victime plus rapidement. Cette technique demanderait par contre une résolution de la caméra assez élevée car il faut réussir à détecter une victime depuis plusieurs dizaines de mètres de hauteur.
Les techniques actuelles de détection utilisées dans les systèmes de *Search and Rescue*, nécessitent plusieurs capteurs afin de traiter plusieurs informations lors du vol. De plus cette méthode permet de détecter des personnes au sol mais la vérification de la détresse de cette personne n'est pas vraiment traitée dans l'article. [1]. Il faudrait donc réfléchir à un moyen de le faire et nécessiterais du temps pour l'implémenter.
L'article [3], arrive à obtenir de fortes performances dans la détection de personnes dans différentes régions. Cependant, cette méthode a été développée dans le but d'assister un agent humain qui va traiter les images enregistrées mais pas en temps réel. L'algorithme utilisé dans cet article doit être amélioré et optimisé afin d'être équipé sur un drone et ainsi permettre un traitement en temps réel.
Dans le cadre des techniques pour l'atterrissage du drone, l'article [4] repose sur la nécessité de disposer d'une zone de marquage afin d'amorcer la phase de descente du drone. Ce qui est en pratique impossible dans le cas d'une opération de recherche et de sauvetage. L'avantage de cette méthode, exclusivement basée sur les images des caméras permet de s'affranchir des erreurs issues des coordonnées GPS pouvant fausser la précision de la descente de l'avion.
L'article [2] quand à lui, utilise une technique de vision qui est associée à l'utilisation des coordonnées GPS afin d'effectuer la descente. Ces coordonnées ne sont pas toujours disponibles ou exactes dans les zones reculées ou quand les conditions météorologiques ne sont pas bonnes. La personne en détresse devra aussi faire face au drone, afin que celui-ci puisse calculer la zone proche de la personne pour se positionner. Cette solution s'est avérée efficace dans différents environnements.

3.1.2 Survol à basses altitudes

Le vol à basse altitude, qui est une solution déjà développée par SII, permet de gagner du temps de développement. Par contre la recherche de la victime sera sûrement plus fastidieuse. Cette solution souffre notamment des difficultés inhérentes à la détection des personnes. En effet, l'article [19], montre que la distance et l'angle de vue depuis le drone influence les performances de la détection de personnes. Ces limites peuvent être surmontées en augmentant le modèle de reconnaissance de personnes par des techniques 3D.
Un autre facteur affectant les performances est la vitesse et le comportement de vol du drone. Ce facteur peut être compensé en paramétrant correctement la caméra du drone.
L'article [6], quant à lui, présente la limite d'avoir été testé sur des images prises sur des caméras monoculaires immobiles. Cette méthode n'est pas adaptée pour des images prises d'un point de vue aérien.

L'inconvénient commun de toutes ces solutions, est que celles-ci n'ont pas été développées pour le sauvetage de personnes en pleine nuit.
Ainsi nous ferons le choix d'utiliser le vol à basses altitudes pour les raisons évoquées ci-dessus. Nous garderons à l'esprit pour le futur qu'une recherche à hautes altitudes serait sûrement plus adaptée aux

activités de secourismes.

3.2 Chatbot pour recueillir ou donner des informations

3.2.1 Modèle Rasa

Comme dit précédemment, Rasa fournit un traitement automatique du langage naturel(NLP) open source pour transformer les messages de l' utilisateur en intentions et entités que les chatbots comprennent. Basé sur des bibliothèques d'apprentissage automatique telles que Tensorflow et spaCy. Rappellons d'abord ce que NLP signifie :
Le traitement automatique du langage naturel (ou bien Natural language processing en anglais) est une catégorie du machine learning qui analyse un texte et le transforme en données structurées. Le NLU (La compréhension du langage naturel) est une sous branche du NLP qui dépasse la conversion du texte en interprétation par classification d'intention en fonction du contexte et du contenu du message.

Dans le monde réel, les messages des utilisateurs peuvent être imprévisibles et complexes, et un message d'utilisateur ne peut pas toujours être mappé à une seule intention. Rasa Open Source est équipée pour gérer plusieurs intentions dans un seul message, reflétant la façon dont les utilisateurs parlent vraiment. Le moteur NLU de Rasa peut distinguer plusieurs objectifs utilisateur, de sorte que l'assistant virtuel réagit naturellement et de manière appropriée, même à des entrées complexes.

3.2.2 Modèle de langage pour les Question Réponses

En ce qui concerne la langue française, [20] a montré sur diverses tâches que leur modèle, FlauBERT, offrait un panel de performances équivalentes à celles de CamemBERT [17], soulignant qui plus est la complémentarité des deux modèles sur des tâches d'analyse syntaxique.
Sachant que ces modèles ont été entraînes sur des données *in fine* différentes bien que d'origine similaire (avec un filtrage plus intense et l'utilisation d'un équivalent francophone du Bookcorpus dans un cas, un filtrage principalement sur le bruit et l'identification de la langue cible dans l'autre), il est pertinent de s'interroger sur l'impact qu'ont les données de pré-entraînement, tant en termes de taille que de type de données, sur les performances des modèles de langue neuronaux contextuels.
D'autres paramètres sont d'importance, en particulier la stratégie de masking utilisée *(subword ou whole-word ?)* et le nombre de couches et de têtes d'attention (modèle Base ou Large ?).

3.2.3 Sur-génération de données d'entraînement du modèle de langage

En ce qui concerne la première approche, Wu et al. [21] ont utilisé un modèle de classification basé sur le « Rough Set » ou « ensemble approximatif » (Pawlak, 1982) et la technique de l'« Ensemble Learning » ou « apprentissage par ensembles » (Ditterrich, 1997) pour prendre une décision. Pour chaque forum, des ensembles de classificateurs approximatifs sont d'abord construits et entraînés. Ils classifient ensuite les réponses du forum et sélectionnent les messages qui y sont liés dans la base de données. La figure 3.1 résume son fonctionnement :

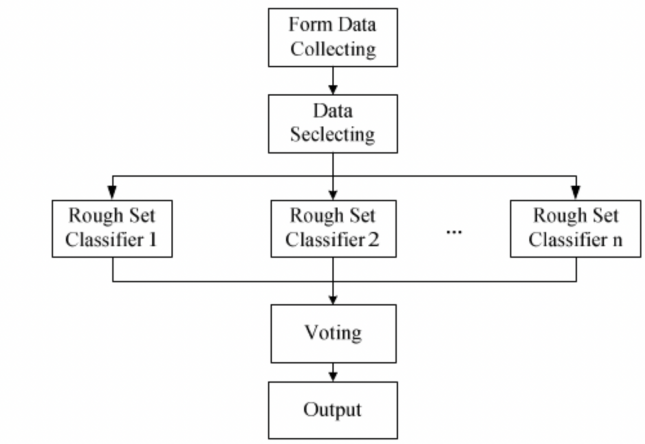

FIGURE 3.1 – Système d'extraction automatique de données

Ils obtiennent de bons résultats mais estiment que beaucoup d'éléments peuvent affecter la qualité des réponses stockées. Ils considèrent alors seulement les caractéristiques structurelles ou propres au contenu, plus à même de permettre une reconnaissance efficace des séquences pertinentes. Ils caractérisent également les difficultés inhérentes à l'analyse de forums en ligne : des formats et des styles de forum différents ou des réponses d'utilisateurs qui ne citent pas le message auquel ils répondent.

Quant à la génération de paraphrases, plusieurs approches sont utilisés :
— Approches classiques (approche par règles [22], par interlingua [23], par thesaurus [24] [25] [26])
— Statistical machine translation (SMT) [27]
— Méthode non supervisée [28]
— Sequence to Sequence (seq2seq) [29]
— Statistical paraphrase generation (SPG) [30]
— Autres méthodes [29]

Chapitre 4

Choix et implémentations

4.1 Positionnement face à une personne

L'implémentation de l'algorithme de positionnement automatique du drone face à une victime se fera en utilisant le langage de programmation Python. Le module PyParrot permettra de contrôler le drone (disponible sur `https://pyparrot.readthedocs.io/en/latest/#`). Ce module permet d'accéder aux informations fournies par les capteurs du Bebop 2 et aussi de positionner le drone dans l'espace selon toutes les directions (avancer, reculer, monter, descendre, tourner, ...). Le contrôle de la caméra est également possible pour changer l'angle de vue par exemple.

En raison de la crise sanitaire, l'accès au drone pour les tests est impossible. Le développement des

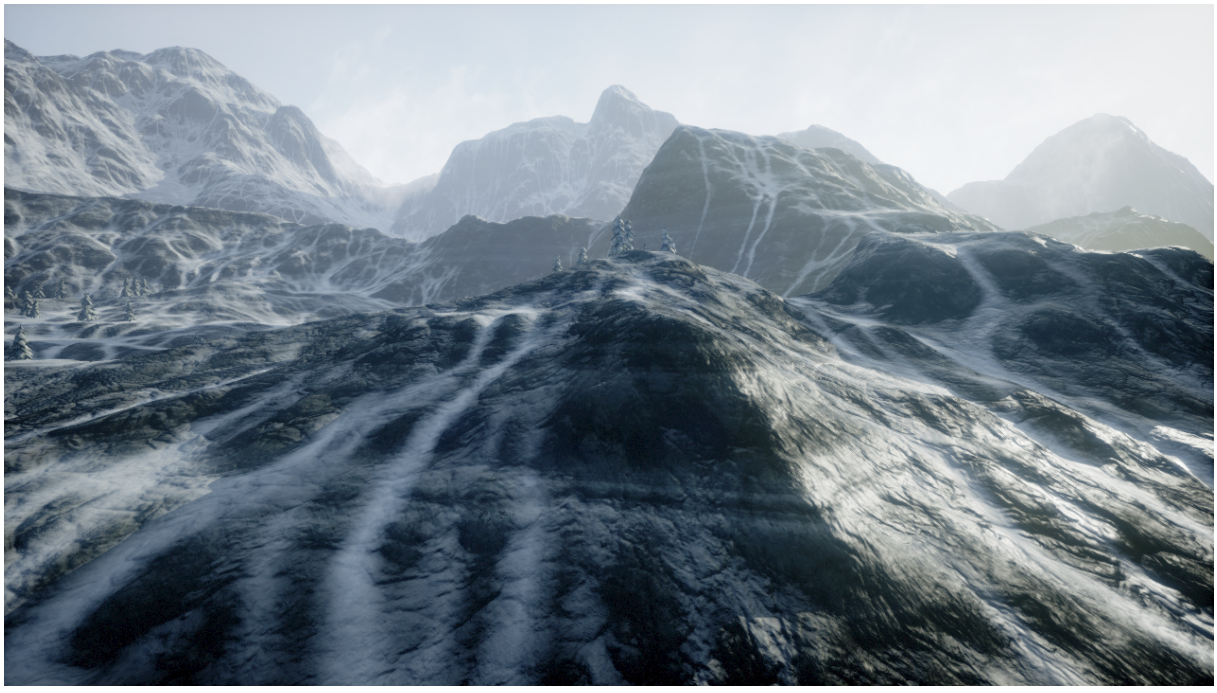

FIGURE 4.1 – Image de Mountains Landscape (Unreal Engine)

algorithmes de positionnement se fera donc à l'aide d'un environnement de simulation : **Unreal Engine**. Pour piloter un drone dans cet environnement simulé nous le couplerons avec AirSim [31] qui est une API de Microsoft. Le codage s'effectuera ensuite en Python à l'aide du package AirSim (lui aussi de Microsoft).

Pour coller au contexte, et c'est là un avantage d'utiliser un simulateur, nous nous placerons dans un environnement de montagne (voir 4.1) : Mountains Landscape. Il faut aussi ajouter un personnage en détresse dans ce décor que le drone puisse détecter avec sa caméra. Adobe à mis en ligne des personnages ainsi que des mouvements à affecter à ces personnages sur le site `www.mixamo.com`.

La détection des victimes se fera à l'aide de *YOLO*, en adaptant évidement les classes de détection à ce qui nous intéresse nous : les personnes. Car *YOLO* est capable de détecter un très grand nombre de classes (voitures, animaux, objets du quotidien, ...). Pour l'estimation de la distance, la solution qui

parait la plus facile à implémenter semble être DisNet car la structure du réseau est simple, et les données d'entraînement sont disponibles en ligne (`https://github.com/guanjianyu/DisNet`).

4.2 Chatbot pour recueillir ou donner des informations

4.2.1 Architecture générale

Lorsque le drone croise un randonneur en difficulté, il lui pose une série de questions dont le scénario est défini dans les règles du serveur Rasa. Si jamais le randonneur a, à son tour, des questions à poser, il questionne le drone, qui reçoit alors un flux audio qu'il envoie à son module de transcription pour le convertir en texte.
Une fois le texte obtenu, le drone envoie la question au serveur Rasa. Si cette question est gérée dans les règles ou scenario du serveur, ce dernier en donne directement la réponse, sinon il la transfert au modèle de langage Camembert "fine-tuné" sur des connaissances en secourisme.

Dans les deux cas, la réponse doit repasser par le module de transcription pour être convertie en format audio avant d'être renvoyée à l'utilisateur.

Une amélioration du système au niveau du serveur Rasa serait de permettre la situation où l'utilisateur ne pose pas vraiment de questions, mais demande à contacter un humain. Une connexion téléphonique devrait alors être établie entre un opérateur et le randonneur. De plus, l'établissement de cette connexion doit se faire en mode asynchrone, pour permettre au drône de continuer à intéragir avec le randonneur entre temps.

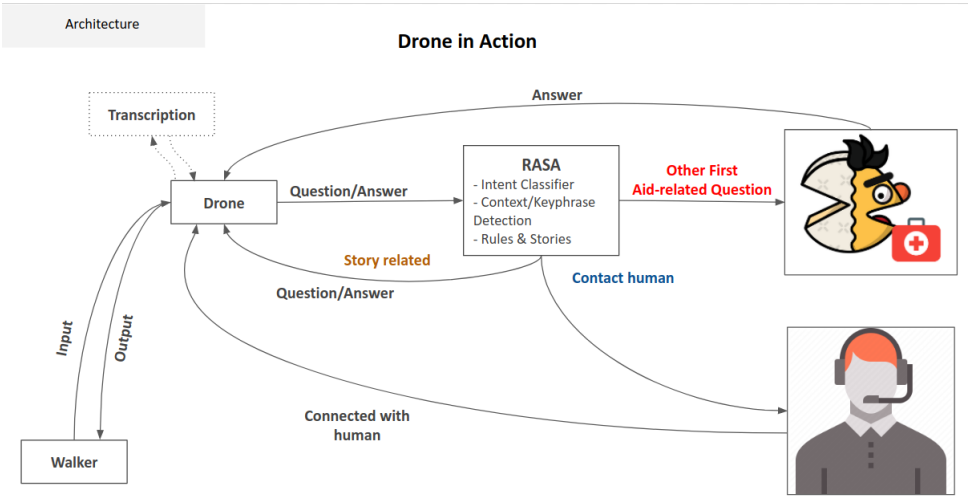

FIGURE 4.2 – Architecture générale

4.2.2 Transcription

Comme mentionné dans le 2.2, nous allons utilisé le transcripteur de LINAGORA utiliser dans LinTO basé sur la boîte à outils open-source Kaldi [8].

4.2.3 Rasa

Après la phase de collecte des données utiles pour notre sujet "aide de personnes en détresse en montagne", viendra la phase d'entraînement de notre modèle via Rasa.

La framework Rasa vient avec multiples fichiers : *yml et python* à modifier afin de personnaliser le chatbot à nos besoins (figurue 4.3).

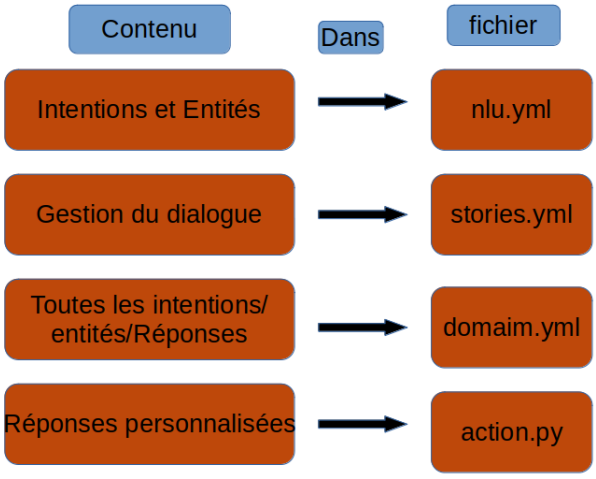

FIGURE 4.3 – Parmi les fichiers à modifier dans la framework Rasa

4.2.4 Modèle de langage pour les Question Réponses

Après réflexion, nous choisissons de partir sur un modèle de langage Camembert pré-entraîné sur les datasets français de questions réponses SQUAD[32] (version Fr) et FQUAD[33]. Ce modèle pré-entraîné est disponible en open source dans la librairie Transformers à l'adresse `shorturl.at/nwCI7`.

Afin d'utiliser ce modèle à bon escient, nous avons besoin de l'entraîner avec des questions-réponses en secourisme.

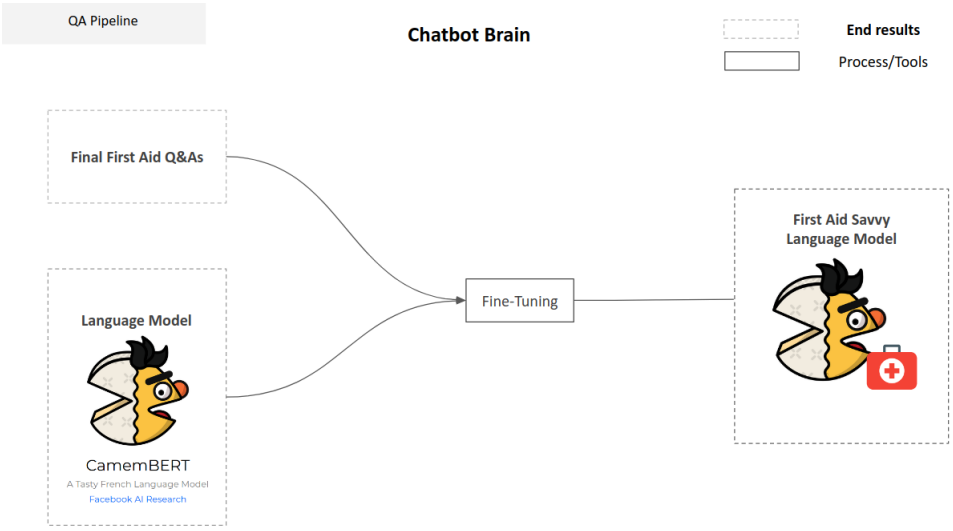

FIGURE 4.4 – Entraînement du modèle

Nous avons donc besoin de récolter de telles données, et pour cela nous avons deux possibilités.

— La première est de faire de l'extraction automatique dans des documents et médias déjà existants, et sous différents formats, à savoir livres, site web, vidéos.

— La deuxième est de constituer nos données à la main par des experts humains.

Ces deux approches aboutissent à un dataset d'une certaine taille, que nous pouvons par la suite augmenter en utilisant des méthodes appropriées décrites dans la section suivante.

4.2.5 Sur-génération de données d'entraînement du modèle de langage

Après réflexions, comme dans l'article [34] nous avons décidé d'implémenter un système basé sur un thesaurus, *Resyf* [35], ce type de méthodes, extrait d'abord tous les synonymes d'un terme avant de sélectionner le plus approprié selon le contexte de la phrase.

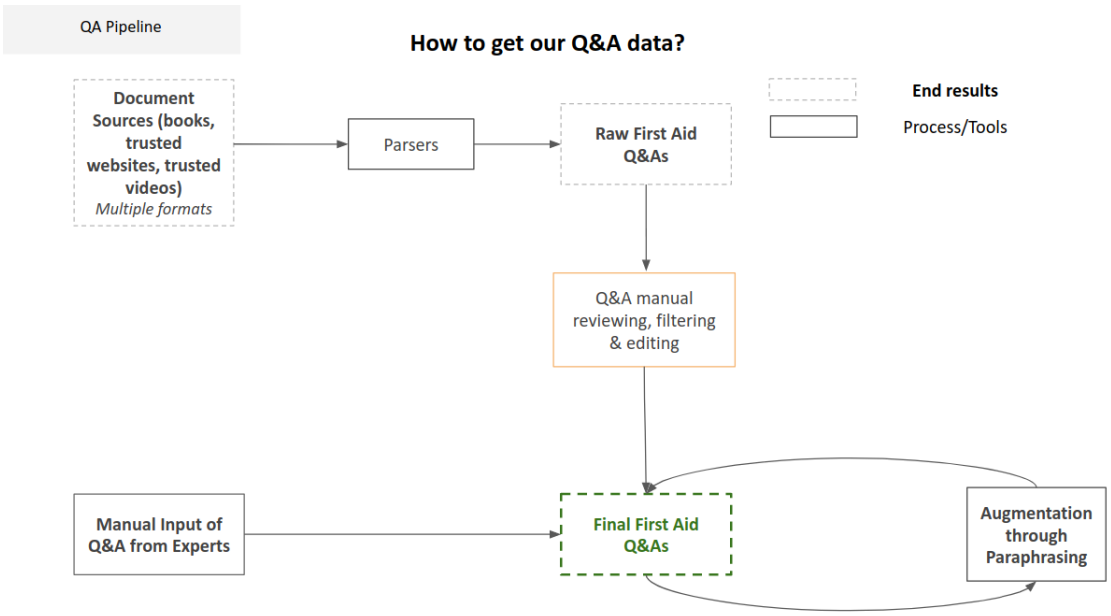

FIGURE 4.5 – Récolte des données d'entraînement

Chapitre 5

Conclusion : Bilan et perspectives

En conclusion, il est à retenir que l'implantation du positionnement du drone est un domaine encore ouvert. Notre idée reposant sur la solution de base de SII et qui s'appuie sur des travaux existants notamment dans le domaine de l'estimation des distances et l'atterrissage d'un drone, n'est pas pleinement adaptée au cas d'usage du Groupe SII.
De nombreuses hypothèses simplificatrices ont été faites notamment sur la hauteur de vol, les conditions de vol et les situations de rencontre d'une personne en détresse. Dans le cadre d'une utilisation moins contraignante, il sera intéressant d'étudier comment réaliser un système qui puisse être performant indépendamment des conditions météorologiques. Un important point d'amélioration serait de pouvoir effectuer des vols à plus haute altitude et aussi pouvoir détecter efficacement des personnes en détresse.
La dernière technique à envisager pour obtenir des résultats encore plus poussées, serait d'étudier la piste de l'apprentissage par renforcement (*Reinforcement Learning*), qui s'avère très prometteuse dans le domaine de la navigation autonome. Cette piste ne fut point explorée dans notre étude en raison du temps limité à notre disposition.
Au niveau de la partie communication, il serait intéressant de tester d'autres transcripteurs, en plus de prendre en compte la distance entre le drone et la personne pour assurer une bonne compréhension. Pour l'outil conversational AI Rasa, il faudra le tester par des secouristes expérimentés qui peuvent mieux simuler les conversations que les victimes pourraient avoir dans différentes situations, ce qui améliorera les performances du Chatbot.
Le dernier point d'amélioration serait d'avoir des codes embarquables aussi performants que les codes non embarquables.

Bibliographie

[1] Jingxuan Sun, Boyang Li, Yifan Jiang, and Chih-yung Wen. A camera-based target detection and positioning uav system for search and rescue (sar) purposes. *Sensors*, 16(11), 2016.

[2] Safadinho David, Ramos João, Ribeiro Roberto, Filipe Vítor, Barroso João, and Pereira António. Uav landing using computer vision techniques for human detection. *Sensors*, 20(3), 2020.

[3] Gotovac Sven, Zelenika Danijel, Marušić Željko, and Božić Štulić Dunja. Visual-based person detection for search-and-rescue with uas : Humans vs. machine learning algorithm. *Remote Sensing*, 12(20), 2020.

[4] Jamie Wubben, Francisco Fabra, Carlos T. Calafate, Tomasz Krzeszowski, Johann M. Marquez-Barja, Juan-Carlos Cano, and Pietro Manzoni. Accurate landing of unmanned aerial vehicles using ground pattern recognition. *Electronics*, 8(12), 2019.

[5] S. Yu, B. Park, and J. Jeong. Posnet : 4x video frame interpolation using position-specific flow. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3503–3511, 2019.

[6] Muhammad AbdulHaseeb, JianyuGuan, Danijela Ristić-Durrant, and Axel Gräser. Disnet : A novel method for distance estimation from monocularcamera. 2018.

[7] J. Bengio. *Reaching new records in speech recognition, https ://www.ibm.com/blogs/watson/2017/03/reaching-new-records-in-speech-recognition/.* Mars 2017.

[8] D. Povey et al. *The Kaldi Speech Recognition Toolkit, Workshop on Automatic Speech Recognition and Understanding.* 2011.

[9] Jean-Pierre Lorré, Isabelle Ferrané, Jorge Francisco Madrigal Diaz, Michalis Vazirgiannis, and Christophe Bourguignat. *LinTO : Assistant vocal open-source respectueux des données personnelles pour les réunions d'entreprise.* hal-02182595, Jul 2019.

[10] Lorena Kodra and Elinda Kajo. Question answering systems : A review on present developments, challenges and trends. *International Journal of Advanced Computer Science and Applications*, 8, 01 2017.

[11] Steven Schockaert, Martine De Cock, and Etienne E. Kerre. Fuzzy constraint based answer validation. In *Proceedings of the Third International Conference on Advances in Web Intelligence*, AWIC'05, page 394–400, Berlin, Heidelberg, 2005. Springer-Verlag.

[12] Dinuksha Kanda Samanage, A. Aneeze, S. Sudheesan, H. Karunaratne, Anupiya Nugaliyadde, and Y. Mallawarrachchi. Advances in natural language question answering : A review, 04 2019.

[13] E. Riloff and M. Thelen. A rule-based question answering system for reading comprehension tests. 2000.

[14] Susanne Humphrey, Aurélie Névéol, Julien Gobeill, Patrick Ruch, Stefan Darmoni, and Allen Browne. Comparing a rule-based versus statistical system for automatic categorization of medline documents according to biomedical specialty. *Journal of the American Society for Information Science and Technology (Print)*, 60 :2530–2539, 12 2009.

[15] S. K. Dwivedi and Vaishali Singh. Research and reviews in question answering system. *Procedia Technology*, 10 :417–424, 2013.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding, 2019.

[17] Louis Martin, Benjamin Muller, Pedro Ortiz Suárez, Yoann Dupont, Laurent Romary, Eric De la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert : a tasty french language model. 11 2019.

[18] A. Gupta, T. Daly, and T. Ban. *Method and system for generating a conversational agent by automatic paraphrase generation based on machine translation.* 2019.

[19] Hwai-Jung Hsu and Kuan-Ta Chen. Face recognition on drones : Issues and limitations. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, DroNet '15, page 39–44, New York, NY, USA, 2015. Association for Computing Machinery.

[20] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert : Unsupervised language model pre-training for french, 2020.

[21] Wu Y., Wang G., Li W., and Li Z. *Automatic chatbot knowledge acquisition from online forum via rough set and ensemble learning.* 2008.

[22] Madnani N. and Dorr B. J. *Monolingual machine translationfor paraphrase generation : A survey of data-driven methods.* Computational Linguistics, 2010.

[23] Glass J. R., Polifroni J., and Seneff S. *Multilingual language generation across multiple domains.* In Third International Conference on Spoken Language Processing, 1994.

[24] Bolshakov I. A. and Gelbukh A. *Synonymous paraphrasing using word-net and internet.* In International Conference on Application of Natural Lan-guage to Information Systems, 2004.

[25] Kauchak D. and Barzilay R. *Paraphrasing for automatic evaluation.* In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, 2006.

[26] Hassan S., Csomai A., Banea C., Sinha R., and Mihalcea R. *Unt :Subfinder : Combining knowledge sources for automatic lexical substitution.* InProceedings of the Fourth International Workshop on Semantic Evaluations(SemEval-2007), 2007.

[27] Quirk C., Brockett C., and Dolan W. *Monolingual machine translationfor paraphrase generation.* In Proceedings of the 2004 conference on empiricalmethods in natural language processing, 2004.

[28] R. Barzilay and L. Lee. *Learning to paraphrase : an unsupervised ap-proach using multiple-sequence alignment*, volume 1. In Proceedings of the 2003 Conference of the North American Chapter of the Association for ComputationalLinguistics on Human Language Technology, 2003.

[29] Li Z., Jiang X., Shang L., and Li H. *Paraphrase generation with deep reinforcement learning.* arXiv :1711.00279, 2017.

[30] Zhao S., X. Lan, T. Liu, and Li S. *Application-driven statistical paraphrase generation.* InProceedings of the Joint Conference of the 47thAnnual Meeting of the ACL and the 4th International Joint Conference onNatural Language Processing of the AFNLP, 2009.

[31] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim : High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[32] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad : 100,000+ questions for machine comprehension of text, 2016.

[33] Martin d'Hoffschmidt, Wacim Belblidia, Tom Brendlé, Quentin Heinrich, and Maxime Vidal. Fquad : French question answering dataset, 2020.

[34] Cheramy Jean. *La génération de paraphrases de données d'entraînement pour une meilleure classification des intents d'un chatbot créé avec Rasa.* Faculté de philosophie, arts et lettres, Université catholique de Louvain, Prom. : François, Thomas. http ://hdl.handle.net/2078.1/thesis :21479, 2019.

[35] Billami, M. B., François T., and Gala N. *Resyf : a french lexicon withranked synonyms.* In Proceedings of the 27th International Conference on Computational Linguistics, 2018.

[36] Nikitina et al. *Crowdsourcing for reminiscence chatbot desig.* 2018.