

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 CADRE THÉORIQUE	5
1.1 Théorie des systèmes de dialogue	5
1.1.1 Dialogues humain-humain	5
1.1.1.1 Acte de langage	5
1.1.1.2 Tours et prise de parole dans une conversation	6
1.1.2 Définition d'un système de dialogue	7
1.1.3 Historique des applications des systèmes de dialogue	8
1.2 Développement des systèmes de dialogue	10
1.2.1 Les différents types de systèmes de dialogue	10
1.2.1.1 Par types de conversation d'humain à machine	10
1.2.1.2 Par degré d'initiative dans la conversation	11
1.2.2 Composants des systèmes de dialogue	12
1.2.2.1 Compréhension du langage naturel	12
1.2.2.2 Gestion de dialogue	14
1.2.2.3 Génération et restitution en langage naturel	16
1.2.3 Méthodes de contrôle du dialogue	17
1.2.3.1 Méthodes par modélisation manuelle	17
1.2.3.2 Méthodes statistiques	19
1.2.3.3 Avantages et Inconvénients	21
1.3 Système de dialogue et prise de rendez-vous	22
1.3.1 L'utilisation des SMS pour la prise de rendez-vous	23
1.3.2 La prise de rendez-vous par échanges en langage naturel	25
1.4 Conclusion	25
CHAPITRE 2 PHASE PRÉLIMINAIRE	27
2.1 Méthodologie et matériel	28
2.2 Récolte des messages	28
2.2.1 Application Web	28
2.2.2 Sélection des participants	29
2.2.3 Choix des moyens de communication	30
2.3 Analyse	31
2.3.1 Prétraitement des messages	31
2.3.2 Annotation d'entités	32
2.3.3 Annotation des expressions temporelles	33
2.3.3.1 Présentation de l'outil HeidelTime	33
2.3.3.2 Annotation des données récoltées	34
2.4 Résultats	35
2.4.1 Phase de récolte des messages	36

2.4.2	Catégorisation des messages	37
2.5	Conclusion	38
2.5.1	Retour sur les objectifs	38
2.5.2	Limites de la méthodologie	39
CHAPITRE 3 MACHINE À ÉTATS		41
3.1	Hypothèses et Définitions	42
3.1.1	Requête de l'utilisateur et rendez-vous	42
3.1.2	La conversation à modéliser	42
3.2	Méthodologie et matériel	43
3.2.1	Méthodologie	43
3.2.2	Choix technologiques	44
3.3	Modélisation	46
3.3.1	Modélisation de la machine à états	46
3.3.2	Modélisation des données	48
3.3.3	Décomposition en modules	49
3.4	Implémentation	49
3.4.1	Compréhension du langage naturel	50
3.4.1.1	Extraction des services	50
3.4.1.2	Extraction des expressions temporelles	52
3.4.2	Gestion du dialogue	52
3.4.2.1	Traitement d'un message	54
3.4.2.2	Développement de la machine à états	55
3.4.3	Génération de la réponse	55
3.4.4	Gestion de la prise de rendez-vous	56
3.5	Conclusion	57
3.5.1	Résultats	57
3.5.2	Avantages et limites de la méthode	59
3.5.2.1	La Gestion de la conversation	59
3.5.2.2	La compréhension de l'énoncé	60
3.5.3	Proposition	60
CHAPITRE 4 EXPLORATION D'UN OUTIL DE CONCEPTION D'INTERFACE CONVERSATIONNELLE		61
4.1	Méthodologie et matériel	62
4.1.1	Méthodologie	62
4.1.2	Choix technologiques	63
4.2	Modélisation	63
4.2.1	Modélisation de la conversation	64
4.2.1.1	Définition des intentions	64
4.2.1.2	Définition des actions	64
4.2.1.3	Modélisation de la machine à états	66
4.2.2	Modélisation des données	68
4.2.3	Décomposition en modules	69

4.3	Implémentation	69
4.3.1	Compréhension du langage naturel	70
4.3.1.1	Création de l'ensemble d'entraînement	70
4.3.1.2	Reconnaissance d'intention dans les messages	71
4.3.2	Gestion de la conversation	71
4.4	Conclusion	72
4.4.1	Résultats	72
4.4.1.1	Exemples d'interactions	72
4.4.1.2	Stratégie d'entreprise	73
4.4.2	Apports et limites	75
CONCLUSION		77
BIBLIOGRAPHIE		81

LISTE DES TABLEAUX

	Page
Tableau 2.1	Détail des profils des participants sélectionnés..... 29
Tableau 2.2	Concept associé aux informations relevées 32
Tableau 3.1	Description des modules 51
Tableau 4.1	Définition des intentions 64
Tableau 4.2	Définition des actions..... 65
Tableau 4.3	Paires intention-action et transitions entre les états 67
Tableau 4.4	Description des modules 70

LISTE DES FIGURES

		Page
Figure 1.1	Architecture ordinaire d'un système de dialogue	15
Figure 1.2	Architecture fondée sur la notion d' <i>état de l'information</i>	16
Figure 2.1	Page de présentation du projet	30
Figure 2.2	Catégorisation des messages	37
Figure 3.1	Diagramme d'activité : processus de prise de rendez-vous	41
Figure 3.2	Architecture de l'application	44
Figure 3.3	Machine à états pour le contrôle de la conversation.....	47
Figure 3.4	Modèle de données	48
Figure 3.5	Décomposition du système en modules.....	50
Figure 3.6	Extraction des expressions temporelles dans un texte.....	53
Figure 3.7	Génération de la réponse	54
Figure 3.8	Exemple 1 : requête fermée	57
Figure 3.9	Exemple 2 : requête semi ouverte et proposition du système	58
Figure 3.10	Exemple 3 : requête semi ouverte et modification des informations	58
Figure 3.11	Exemple 4 : traitement du refus de l'utilisateur	59
Figure 4.1	Architecture de l'application	63
Figure 4.2	Modèle des données	68
Figure 4.3	Décomposition en modules.....	69
Figure 4.4	Ensemble d'exemples de l'intention <i>réserver</i>	71
Figure 4.5	Catégorisation des messages	72
Figure 4.6	Exemple 1 : gestion d'une requête simple	73
Figure 4.7	Exemple 2 : gestion d'un refus	74

Figure 4.8	Exemple 3 : demande de disponibilités	74
------------	---	----

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Interface de Programmation
BDD	Base De Données
CLN	Compréhension du Langage Naturel
ÉTS	École de technologie supérieure
FSM	Machine à états
GRC	Gestion de la Relation Client
IPA	Agent Personnel Intelligent
MDP	Processus de Décision Markovien
MVC	Modèle, Vue et Contrôleur
MVT	Modèle, Vue et Template
POMDP	Processus de Décision Markovien Partiellement Observable
RV	Rendez-vous
SD	Systèmes de Dialogue
SMS	Service de message court
SQL	Langage de Requête Structurée
TALN	Traitement Automatique du Langage Naturel

INTRODUCTION

Avec l'intérêt croissant pour l'intégration d'agents intelligents dans les systèmes informatiques, les interfaces graphiques tendent à être remplacées par des interfaces conversationnelles permettant aux utilisateurs d'interagir avec un système en s'exprimant dans leur langage naturel (Lison et Meena, 2014). Depuis le début des années 2010, le "commerce conversationnel" suscite un nouvel intérêt de la part des entreprises. Ce nouveau terme, introduit par Chris Messina (2015), fait référence à l'utilisation de messages instantanés par les entreprises pour communiquer avec leurs clients et leur proposer leurs services. Elles cherchent à se rapprocher de leurs clients en rendant leurs interactions plus naturelles, intimes et personnalisées. L'utilisation d'applications de clavardage ou des SMS devient stratégique dans cette approche. Cet intérêt s'appuie sur les constats selon lesquels les applications de messagerie instantanées sont très populaires et utilisées quotidiennement par les consommateurs¹ et que le nombre de SMS envoyés ne cesse d'augmenter².

Afin de garder son caractère instantané, la conversation d'humain à humain par l'intermédiaire de ce type de moyen de communication demande la mise à disposition de nombreuses ressources pour répondre aux messages dans les plus brefs délais. C'est pourquoi, l'intégration d'agents conversationnels intelligents, pour automatiser les conversations en langage naturel, est étudiée. Au-delà d'applications particulières réservées à la recherche, l'intégration de ces technologies est devenue possible grâce à l'évolution des méthodes d'intelligence artificielle et l'émergence de nouveaux outils, d'aide à la conception d'applications, à la portée de tous développeurs.

Les recherches en intelligence artificielle et l'implémentation d'agents conversationnels ont commencé dans les années 1960, mais restaient jusqu'à présent limitées à des utilisations spécifiques. En revanche, depuis le début des années 2010, avec l'émergence des téléphones intelligents, les assistants personnels intelligents (IPA) sont de plus en plus appréciés et utilisés par les usagers dans leur vie courante. Ces agents intelligents reconnaissent des requêtes en

1. On pourrait citer, whatsapp, viber, Facebook Messenger, wechat (en Asie), etc.

2. En 2014, il était estimé que, à travers le monde, plus de 350 milliard de SMS envoyés par mois (The Open University, 2014)

langage naturel et réalisent une tâche ou génèrent une réponse. Bien que les performances des algorithmes et des techniques sur lesquelles ils reposent s'améliorent, la conversation automatique en langage naturel est encore restreinte et repose souvent sur l'utilisation d'expressions prédéterminées par le système.

La collaboration avec une entreprise, spécialisée dans le développement d'une suite logicielle pour la gestion de la relation client (GRC), est à l'origine du projet présenté dans ce mémoire. Pour les entreprises, la prise de rendez-vous est un processus important. Cette suite logicielle comprends un système d'ordonnancement pour la gestion de leurs calendriers et une fonctionnalité de personnalisation d'un outils de prise de rendez-vous intégrable dans les sites internet. Ce dernier permet d'automatiser le processus et de rendre les entreprises disponibles en permanence. Le constat est le suivant, bien que les entreprises de services mettent à la disposition de leurs clients des outils de prise de rendez-vous en ligne, elles remarquent que le recours aux appels téléphoniques reste privilégié. De cette manière, le client espère bénéficier d'un rendez-vous plus adapté à ses contraintes, à son profil et à ses besoins. Par conséquent, la charge de travail n'est pas réduite pour les entreprises qui continuent à gérer leurs rendez-vous par téléphone et sur des plages horaires restreintes à leurs heures d'ouverture.

L'objectif de l'entreprise partenaire, à long terme, est donc de concevoir un système automatique de prise de rendez-vous par téléphone. De la même manière qu'elle propose aujourd'hui, un outil de prise de rendez-vous en ligne, elle offrirait un système personnalisable de dialogue vocal ou textuel. Ce dernier serait capable de gérer des conversations permettant à un client d'interagir en langage naturel pour prendre un rendez-vous. L'objet de ce projet s'intéresse au cas particulier de la prise de rendez-vous par SMS.

L'utilisation des SMS pour interagir avec un utilisateur n'est pas récente. La littérature nous a permis d'identifier des recherches visant à utiliser ce moyen de communication dans le cadre de la prise de rendez-vous. Souvent limitée par la compréhension du langage naturel, la conversation est restreinte à des échanges prédéterminés. Les messages textes sont, soit utilisés à des fins de rappels, envoyés d'une entreprise vers un client pour lui rappeler son prochain rendez-

vous, soit dans un contexte de dialogue automatisé, mais où l'utilisateur est contraint d'utiliser des expressions restreintes par la compréhension du système. Les techniques de Traitement Automatique du Langage Naturel (TALN) évoluent et permettent de rendre les échanges plus libres et flexibles. Dans ce type de système, la difficulté ne réside pas seulement dans la compréhension de l'énoncé de l'utilisateur. En effet, une conversation repose sur un ensemble de tours de paroles entre différents interlocuteurs. La gestion d'une conversation repose sur le contrôle du dialogue et en particulier la modélisation d'une stratégie pour déterminer l'action à réaliser par le système (El Asri *et al.*, 2014b). L'intégration d'interfaces conversationnelles entre un système et un utilisateur soulève des difficultés dépendantes du contexte et du type de conversation. Par exemple, dans un contexte de prise de rendez-vous, Busemann *et al.* (2011) étudient la construction collaborative du rendez-vous entre les différentes parties impliquées pendant les tours de paroles de la conversation.

Le présent projet est une étude exploratoire de l'intégration d'une interface conversationnelle dans un système de prise de rendez-vous par SMS.

Nos objectifs étaient les suivants :

- 1 Identifier l'effet du changement du moyen de communication sur la gestion de la prise de rendez-vous ;
- 2 Dégager les caractéristiques particulières d'une conversation pour la prise de rendez-vous.

L'étude a été appuyée par le développement d'un prototype de système conversationnel pour la prise de rendez-vous pour lequel nous avons identifié trois objectifs :

- 1 La compréhension des données pour déterminer une catégorisation et une modélisation conceptuelle des messages des utilisateurs ;
- 2 La détermination d'une méthode pour la gestion des échanges ;
- 3 L'utilisation de méthodes de représentations sémantiques utilisées dans les systèmes de dialogue (SD).

Le premier chapitre présente le cadre théorique de l'étude. Tout d'abord, une définition théorique des systèmes de dialogue homme-machine et les méthodes de développement associées y sont décrites. Puis, les différentes hypothèses sur la prise de rendez-vous et l'utilisation du langage naturel y sont énoncées. **Le deuxième chapitre** présente la phase préliminaire composée d'une phase de récolte et d'une phase d'analyse de données. **Le troisième et le quatrième chapitre** décrivent les deux approches étudiées pour concevoir un prototype de système de prise de rendez-vous en langage naturel. **La conclusion** revient sur les objectifs initiaux portant sur l'utilisation d'échanges en langage naturel et son influence sur le processus de prise de rendez-vous. Puis, elle propose des axes de recherche et de développement de la poursuite du projet de l'entreprise partenaire.

CHAPITRE 1

CADRE THÉORIQUE

Ce chapitre vise à présenter le cadre théorique de l'étude. La première section définit théoriquement les systèmes de dialogue. La seconde section détaille les différentes méthodes de développement, leurs avantages et inconvénients, ainsi que leur application à notre contexte. La dernière section présente différentes études utilisant le langage naturel et en particulier les SMS pour la prise de rendez-vous entre un humain et une machine.

1.1 Théorie des systèmes de dialogue

Dans cette section, nous proposons une définition des systèmes de dialogue (SD). Ces systèmes visent à reproduire des conversations d'humain à humain. Nous présentons les propriétés des conversations entre humains nécessaires à la compréhension des choix effectués dans la conception de SD (Jurafsky et Martin, 2014). Nous définissons les SD en général puis ceux appliqués aux échanges vocaux et textuels.

1.1.1 Dialogues humain-humain

Des éléments de conception des SD reposent sur des études philosophiques et sociologiques relatives aux interactions entre humains dans le cadre des conversations. Dans cette section nous définissons les concepts d'« actes de langage » et de « tours de parole » dans les conversations.

1.1.1.1 Acte de langage

L'acte de langage donne une interprétation de haut niveau d'un énoncé en langage naturel. Cette notion ¹ a été introduite en 1962 par le philosophe John L. Austin (1962) : « Dire c'est faire » ². Il propose d'associer l'énoncé d'un intervenant dans un dialogue à un acte pouvant avoir un

1. On retrouvera aussi, dans la littérature et dans les études du langage, les termes : actes de langage ou acte de parole.

2. Traduit de l'anglais « Do things with words »

effet dans une conversation, sur les autres interlocuteurs et sur l'environnement en général. Considéré comme un acte associé au langage, il est aussi composé d'un but, un prérequis, un corps et un effet. Il oppose d'abord, l'acte constatif à l'acte performatif. L'acte constatif a une fonction de description : on peut associer à ces énoncés une valeur de vérité. Alors que l'acte performatif décrit l'énoncé comme une action qui peut réussir ou échouer si elle est exécutée.

Par la suite, John L. Austin (1975) propose une classification approfondie décrivant un énoncé selon trois types d'actes : locutoires, illocutoires et perlocutoires. L'acte locutoire est le simple fait de dire quelque chose. L'acte illocutoire représente une intention (verdictive, promissive, exercitive, comportementative ou expositive) exprimée par le locuteur. L'acte est dit perlocutoire, si son énonciation par le locuteur a produit un effet sur son interlocuteur ou sur l'environnement³. Finalement, un acte de langage donne une représentation de l'intention d'un locuteur dans une conversation indépendamment de son contenu sémantique. On décrit en général un acte de langage par sa fonction illocutoire.

John R. Searle (1969) reprend ces travaux et approfondit le concept illocutoire, en produisant une classification selon cinq types d'actes de langages : assertifs (expression de l'engagement du locuteur), directifs (expression de la volonté ou de désir que l'interlocuteur réalise quelque chose), promissifs (engagement dans une action), expressifs (expression d'un état psychologique) ou déclaratifs (affirmation d'une réalité sur le monde).

1.1.1.2 Tours et prise de parole dans une conversation

Un dialogue est par définition un ensemble d'échanges tenus entre des individus. La conversation est articulée selon des tours de parole entre les intervenants. Les études s'orientent selon deux axes : la détermination d'une structure générique de la conversation et la proposition de solution pour déterminer la fin d'un tour de parole. La discipline de l'Analyse Conversationnelle (AC), dont Harvey Sacks est le fondateur, vise à étudier les comportements des interac-

3. Par exemple, dans l'énoncé « Peux-tu me prêter ton vélo », le contenu locutoire est le fait que l'interlocuteur ait un vélo (contenu). Alors que la fonction illocutoire est le fait de vouloir emprunter un vélo. Finalement, la fonction perlocutoire peut produire l'effet que l'interlocuteur accepte et prête son vélo (effet sur le récepteur).

tions dans une conversation. Contrairement aux idées présentées jusqu'alors, l'AC estime que ces tours de parole sont dirigés et ordonnés. Dans leurs travaux, Harvey Sacks, Emanuel A. Schegloff et Gail Jefferson, les trois pionniers du mouvement, décrivent un ensemble de règles génériques et systémiques régissant les conversations entre individus (Sacks *et al.*, 1974). Les différentes règles et critères présentés permettent de décider qui doit ou peut parler par la suite. La détermination de la fin d'un tour de parole et du passage de la parole est étudiée selon les indices suivants :

- les silences : l'étude de la durée d'un silence à la suite d'un énoncé ;
- l'intonation de la voix : donne la parole à quelqu'un ;
- le langage corporel : désigner la personne à qui la parole est donnée.

Ces notions sont centrales dans la conception de SD. En particulier dans ceux utilisant la voix, où la détermination de la fin d'un tour de parole est une problématique étudiée dans les systèmes de reconnaissance vocale (Raux et Eskenazi, 2012).

1.1.2 Définition d'un système de dialogue

Un système de dialogue est un programme informatique capable de converser de manière naturelle, cohérente et structurée avec un humain (McTear, 2002). Il existe différents modes d'interaction : les interfaces graphiques, la voix, le texte, les expressions corporelles ou les expressions du visage. Ces modes peuvent aussi être combinés (systèmes multimodes). Dans le cas particulier des échanges textuels ou vocaux, la littérature utilise différents termes pour définir les systèmes : agents conversationnels (Jurafsky et Martin, 2014), interfaces conversationnelles (M. Smart, 2016) ou les systèmes de dialogue parlé⁴ (McTear, 2002) pour les échanges vocaux.

D'un point de vue théorique, l'ambition principale des SD est de reproduire des dialogues d'humain à humain pour permettre à un utilisateur de communiquer plus librement avec un système. Les interfaces graphiques actuelles imposent à l'utilisateur une interaction et des échanges dé-

4. Dans notre projet, nous nous sommes concentrés sur la définition des systèmes de dialogue en langage naturel utilisant la voix ou le texte comme moyen d'interaction qui seront désignés par le terme SD dans ce mémoire.

terminés qui impliquent l'appropriation par l'utilisateur des fonctions du système. Or, le langage naturel ne nécessite pas d'apprentissage au préalable et permet un échange plus souple (Lison et Meena, 2014).

Cependant dans l'application, permettre à un utilisateur de s'adresser librement et de se faire comprendre par un système relève de nombreuses problématiques concernant les tâches de : reconnaissance de l'entrée dépendante du mode considéré, compréhension du langage naturel, gestion de la conversation, génération d'une réponse et restitution à l'utilisateur dans le mode choisi.

1.1.3 Historique des applications des systèmes de dialogue

Les différentes applications des SD en langage naturel ont évolué en fonction des motivations et des moyens technologiques à disposition.

À l'origine, dans les années 1960, alors que l'intérêt se porte sur l'intelligence artificielle, l'une des motivations est de créer des systèmes capables de passer le test dit de « Turing ». Ce test, proposé en 1950 par Alan Turing (Turing, 1950) a été conçu pour évaluer les systèmes intelligents. Il confronte le système à un utilisateur : si ce dernier n'est pas capable de déterminer qu'il est en train de dialoguer avec un système informatique alors le test est réussi. Les premiers agents conversationnels visaient à reproduire des comportements humains. Le système ELIZA développé en 1966 (Weizenbaum, 1966) vise à reproduire le comportement d'un psychothérapeute et a inspiré de nombreux autres développements par la suite. Le processus de génération d'une réponse repose sur la reconnaissance de mots clés dans les textes et l'application d'un ensemble de règles qui associent un énoncé à une réponse. Le système Parry (Colby, 1981), quant à lui, reproduit le comportement d'un schizophrène. La gestion de la conversation est améliorée par l'intégration de variables représentant l'état interne du système. En 1995, Richard S. Wallace, motivé par les lacunes du système ELIZA, développe l'agent conversationnel A.L.I.C.E. basé sur le langage d'annotation AIML (Wallace, 2003). Il a remporté à trois reprises le Prix Loebner récompensant les systèmes intelligents à partir de leur résultat

au test de Turing. Cependant, le traitement du langage naturel et les mécanismes d'intelligence artificielle sont limités à la reconnaissance de mots clés ou d'expressions directement liées aux réponses du système.

Depuis les années 1970, l'évolution des SD est fortement liée au développement des études sur les conversations d'humain à humain⁵ qui améliorent les méthodes de gestion du dialogue. Dès lors, les SD sont intégrés dans les téléphones, les ordinateurs, les robots, les voitures, etc. Par exemple, on retrouve des applications des SD pour la planification de voyage (Aust *et al.*, 1995), la réservation de vols (Hemphill *et al.*, 1990), la redirection des appels dans les centres téléphoniques (Gorin *et al.*, 1997) ou encore l'éducation (Litman et Silliman, 2004). En 1999, un langage d'annotation VoiceXML est normalisé par AT&T, Lucent, Motorola et IBM et permet de créer les premières applications commerciales de SD par la voix (VoiceXML, 2016). Ces

Depuis la fin des années 1990, l'émergence de l'Internet génère de larges ensembles de données. Le langage VoiceXML devient un standard World Wide Web Consortium (W3C) qui l'utilise pour des développements d'applications vocales sur le Web (World Wide Web Consortium, 2007). De plus, l'émergence des appareils mobiles marque le début des assistants virtuels. Ces agents intelligents sont dotés d'un ensemble de connaissances et sont capables d'interagir avec un utilisateur pour réaliser des tâches, faire des recommandations, reconnaître des commandes, etc. En 2011, SIRI (le premier assistant personnel intégré dans les téléphones mobiles) est le résultat de recherches menées par le centre SRI dans le cadre du projet CALO (SRI, 2016). Par la suite, d'autres fournisseurs développent leurs propres applications commerciales comme Google (Google Now), Samsung (S Voice) ou Amazon (Alexa). En 2015, dans le cadre d'applications académiques, l'université du Michigan développe le premier assistant personnel intelligent libre : Sirius (Hauswald *et al.*, 2015). Ces assistants personnels marquent un changement par rapport aux développements précédents car ils s'intègrent largement dans le quotidien des utilisateurs. Dans un premier temps, ils sont installés dans les téléphones mobiles

5. Nous avons présenté des études de philosophes et sociologues à l'origine des études des conversations dans la section 1.1.1

à des fins de commandes vocales. Puis, la tendance du « commerce conversationnel » élargit leur application en les intégrant dans différents types d'applications de clavardage.

1.2 Développement des systèmes de dialogue

Les SD peuvent avoir des architectures différentes selon les moyens d'interactions qu'ils utilisent. Cependant, ils se décomposent toujours selon trois tâches principales (McTear, 2002) :

- la compréhension du langage naturel (CLN) : donner un sens à l'énoncé de l'utilisateur (vocal ou textuel) grâce à des techniques de Traitement Automatique du Langage Naturel (TALN) ;
- la gestion de la conversation : gérer les échanges entre l'utilisateur et la machine pour entretenir un dialogue cohérent ;
- la génération et la restitution en langage naturel : retourner une réponse en langage naturel (vocal ou textuel) à l'utilisateur en respectant les règles grammaticales de la langue.

Dans cette section, nous présentons les méthodes et les approches étudiées pour la conception de chacun de ces composants. Le choix de développement dépend principalement du type de système visé. Nous présentons les types de systèmes que nous avons identifiés et nous détaillons les méthodes de contrôle du dialogue pour lesquelles nous avons porté un intérêt particulier lors du développement de notre prototype.

1.2.1 Les différents types de systèmes de dialogue

La littérature distingue deux types de classification des SD, selon : le type de conversation d'humain à humain reproduit ou le degré d'initiative de l'utilisateur dans la conversation.

1.2.1.1 Par types de conversation d'humain à machine

Les SD peuvent être classés selon l'objectif visé et la complexité de la conversation qui peut être tenue. Par exemple, dans les systèmes de commandes, le dialogue est restreint à l'exécution d'une action et à des échanges unidirectionnels (de l'utilisateur vers le système). Les systèmes

de questions-réponses, quant à eux, effectuent des recherches en base de données (BDD) à partir d'une requête énoncée par l'utilisateur. Les échanges peuvent être bidirectionnels car le système pose des questions pour compléter la requête. Les systèmes orientés vers un but s'opposent aux systèmes sans but (Banchs et Li, 2012). Les premiers modélisent des conversations ayant un but précis, comme la prise de rendez-vous (El Asri *et al.*, 2014b), la réservation de trains (Aust *et al.*, 1995), les assistants personnels (Hauswald *et al.*, 2015), l'éducation (Litman et Silliman, 2004), etc. Les seconds, permettent de maintenir une conversation sur des sujets divers (Nio *et al.*, 2014) comme dans le système ELIZA (Weizenbaum, 1966) ou A.L.I.C.E (Wallace, 2003).

1.2.1.2 Par degré d'initiative dans la conversation

Les SD peuvent être classés selon leur degré d'initiative (Lemeunier, 2000). La notion de degré d'initiative est directement liée aux « tours de parole » (*voir* Section 1.1.1.2). Dans une conversation entre un utilisateur et une machine, le degré d'initiative signifie à quel point la conversation est structurée et par qui elle est dirigée (l'utilisateur, le système ou les deux à la fois).

Dans le cas où l'utilisateur est à l'initiative, il mène la conversation (c'est ce qu'il va dire qui va diriger la suite du dialogue). Dans le cas contraire, c'est le système qui mène la conversation et guide l'utilisateur.

Dans le cas d'une initiative mixte, l'utilisateur et le système peuvent intervenir dans la conversation quand ils le désirent, ce qui permet de représenter des conversations plus ouvertes et naturelles. Dans ce cas, la gestion des tours et la détermination de la fin d'un tour de parole complexifie la tâche de contrôle du dialogue.

Les types de systèmes présentés dans cette section mettent en évidence la notion de gestion et de modélisation d'une conversation qui dépendent du type de conversation que l'on souhaite mener. Par la suite, nous présentons des méthodes et approches pour chacun des composants d'un SD et plus particulièrement pour le contrôle du flux du dialogue.

1.2.2 Composants des systèmes de dialogue

1.2.2.1 Compréhension du langage naturel

Le rôle du composant de compréhension du langage naturel est de donner un sens à un énoncé (textuel ou vocal) en déterminant une représentation sémantique, permettant au module de gestion du dialogue de prendre une décision sur la suite de la conversation (Lison et Meena, 2014). La compréhension du langage naturel est une discipline particulière du TALN alliant les domaines de l'informatique, de l'intelligence artificielle et de la linguistique. Le traitement s'opère sur quatre niveaux : morphologique, syntaxique, sémantique et pragmatique (Ratté, 2015).

La CLN dans les systèmes de dialogue par la voix inclut un composant de reconnaissance automatique de la parole (ASR). Un signal vocal est traduit en un ensemble de symboles qui forment des mots. L'ensemble de mots est reconstruit pour former l'énoncé final. L'ASR retourne plusieurs hypothèses d'énoncés finaux. Ce traitement repose sur trois modèles statistiques. Le modèle acoustique estime la probabilité d'observer un vecteur acoustique étant donné un signal vocal. Le modèle de la prononciation estime la probabilité d'observer un mot étant donné une prononciation. Enfin, le modèle de langage donne la probabilité d'une suite de mots dans le langage ciblé. En général, l'ASR retourne un ensemble d'hypothèses de suite de mots associées à une probabilité (McTear, 2002).

Les méthodes utilisées pour la CLN se divisent selon trois approches (Stoyanchev, 2015; Meurs, 2009) : linguistique (à base de règles), stochastique (modèles probabilistes) ou par classification. De Mori *et al.* (2008) proposent une revue complète des approches utilisées dans le cadre particulier des SD par la voix. Cependant, puisque la séquence produite par le synthétiseur vocal est considérée comme un ensemble de mots alors le processus d'analyse sémantique appliqué est le même que pour les énoncés écrits.

1.2.2.1.1 Approche linguistique

L'approche linguistique requiert une analyse syntaxique et sémantique d'un énoncé. D'abord, l'analyseur syntaxique produit un arbre syntaxique dans lequel on associe à chaque mot ou ensemble de mots, une catégorie (verbe, nom, déterminant, groupe nominal, etc.). Puis, à partir d'un ensemble de règles, un rôle sémantique est associé à des composants de la phrase (Allen et Perrault, 1980). Cependant, la particularité des dialogues implique d'effectuer un traitement approfondi pour considérer l'ensemble des échanges déjà effectués (Allen *et al.*, 2007).

Une représentation sémantique des concepts et des associations extraites de ces analyses repose sur le formalisme des cadres sémantiques (Fillmore, 1985). Un énoncé est représenté par un prédicat et une liste d'arguments. Cette approche est pertinente dans la CLN dans les SD où une conversation est un ensemble de cadres dans lesquels les valeurs des arguments sont complétées au cours de la conversation (Meurs, 2009). Plus particulièrement, un cadre peut identifier un acte de langage⁶ dans lequel les arguments sont des attributs de l'acte.

Cependant, la phase de synthèse vocale produit des phrases qui ne sont pas grammaticalement correctes. D'une part, la cohérence de la phrase peut être erronée, d'autre part, la ponctuation n'est pas toujours fournie. La même observation est faite dans des échanges textuels où les phrases de l'utilisateur ne sont pas toujours correctement formées.

Les approches présentées précédemment reposent sur des structures grammaticales robustes et sensibles aux erreurs. C'est pourquoi les travaux se sont rapidement orientés vers des approches reposant sur des grammaires stochastiques. Par exemple, l'utilisation de grammaires probabilistes (Seneff, 1992) associe des probabilités calculées à partir d'un ensemble d'apprentissages aux règles de grammaire. Cependant, elles produisent des systèmes dépendants du domaine.

6. Concept extrait des études en philosophie et sociologie présentée dans la section 1.1.1

1.2.2.1.2 Approche stochastique

Les approches stochastiques se fondent sur des modèles probabilistes et l'apprentissage de paramètres basés sur des ensembles de données. On peut citer par exemple, l'ensemble de données de référence ATIS pour la réservation de vols (Hemphill *et al.*, 1990) ou MEDIA pour l'information touristique (Bonneau-Maynard *et al.*, 2009). Dans ce cas, l'analyse sémantique de l'énoncé est directement réalisée à partir du signal acoustique. On cherche à déterminer une séquence de concepts étant donné une séquence acoustique. Différentes approches ont été proposées, d'abord, les Chaînes de Markov Cachées (Pieraccini *et al.*, 1991), puis inspirées par ce modèle, les HUM (Hidden Understanding Models, (Miller *et al.*, 1994), les arbres de classifications sémantiques (Kuhn et De Mori, 1995), des modèles de vecteurs à états cachés (He et Young, 2003) ou encore l'utilisation de réseaux bayésiens dynamiques (Meurs *et al.*, 2009).

1.2.2.1.3 Approche par classification

La tâche de CLN peut aussi être considérée comme une tâche de classification (De Mori *et al.*, 2008). Ces approches classifient les énoncé par des intentions prédéterminées dans le système. Des caractéristiques sont extraites des phrases, en particulier les *n-grams* (une séquence de *n* mots dans le cadre des textes) et des entités pertinentes et prédéterminées pour la classification. Des corpus de références (Godfrey *et al.*, 1992), ont été annotés pour réaliser cette tâche de classification. Différentes méthodes sont utilisées incluant des approches bayésiennes HMM (Stolcke *et al.*, 2000), SVM (Ribeiro *et al.*, 2015) ou des réseaux de neurones à plusieurs couches.

1.2.2.2 Gestion de dialogue

Le gestionnaire de dialogue est en charge de conduire la conversation. En fonction de l'état de la conversation, des hypothèses d'actes de langage et des informations extraites par le CLN, il détermine l'action à entreprendre et la réponse à retourner. Il est aussi responsable de la mise

à jour des informations et de l'état interne du système. La section 1.2.3 présente différentes approches de contrôle de dialogue, ainsi que leur contexte d'application.

Pour concevoir un système complet de bout en bout qui intègre chaque composant pour le traitement de l'énoncé en entrée jusqu'à la production d'une réponse en sortie, chaque application peut définir son architecture. Certaines alignent simplement les composants (*voir* Figure 1.1), la sortie d'un module est l'entrée du suivant. L'information est traitée de manière séquentielle et le gestionnaire de dialogue est le module central du système (Lison et Meena, 2014).

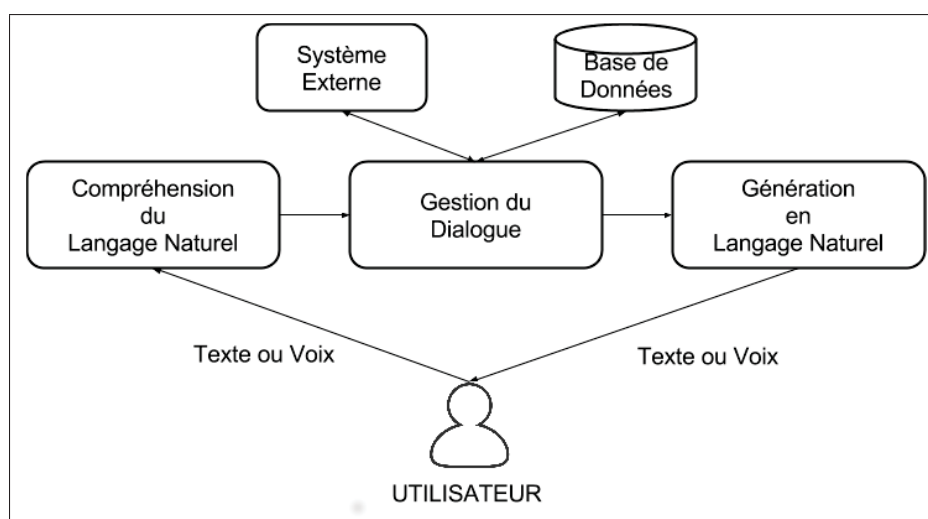


Figure 1.1 Architecture ordinaire d'un système de dialogue

En revanche, Larsson et Traum (2000) proposent une architecture centrée sur l'état de l'information⁷ qui permet de distinguer le traitement de l'information contenue dans la conversation du rôle de gestionnaire de dialogue. Dans ce cas, chaque composant accède et modifie les données indépendamment des autres modules et peut être indépendant du système développé (*voir* Figure 1.2). La conception et la modélisation de la conversation se concentrent sur les données. Il faut d'abord déterminer quelles sont les informations, leurs types (statique ou dynamique) ainsi que leur représentation (sous forme d'acte de dialogue). Puis, déterminer l'ensemble des actes de dialogues, des énoncés entrants et des réponses pouvant être retournées⁸ prises en

7. Traduction de l'anglais *Information State Approach*

8. La notion d'acte de dialogue est définie dans la section 1.1.1

compte par le système. Enfin, les règles de mise à jour et une stratégie (ensemble de règles) pour appliquer ces règles sont déterminées. Le rôle du gestionnaire de dialogue est de sélectionner la stratégie adéquate en fonction de l'état de l'information.

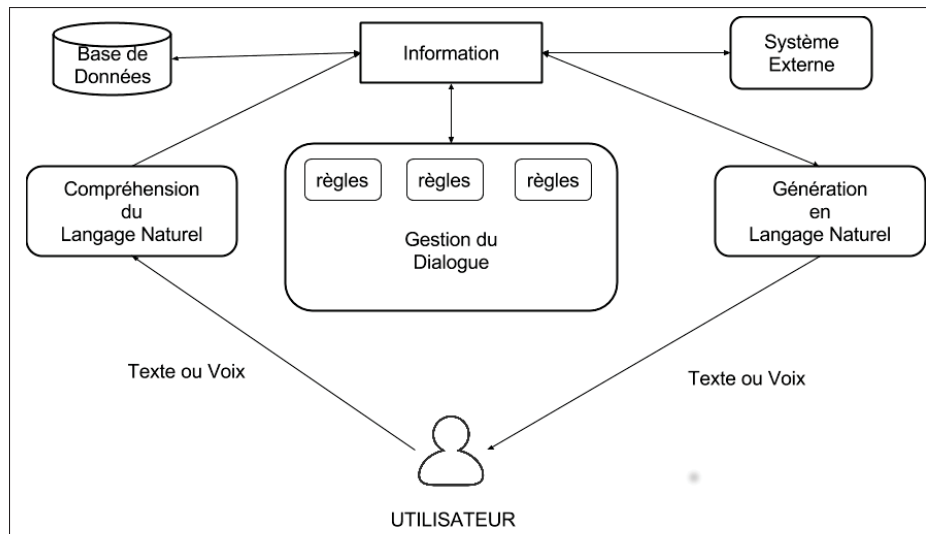


Figure 1.2 Architecture fondée sur la notion d'état de l'information

1.2.2.3 Génération et restitution en langage naturel

La troisième tâche d'un système de dialogue est de retourner une réponse en langage naturel. Les composants à intégrer dépendent du mode d'interaction choisi. Dans le cas des échanges par la voix, un composant de synthèse vocal est utilisé pour retranscrire un texte en signal vocal. Que ce soit pour les échanges vocaux ou écrits, un texte en langage naturel doit être généré.

La génération de texte en langage naturel est la seconde tâche du TALN. Des spécificités sont soulignées quant à la réalisation de cette tâche dans le contexte d'un dialogue (McTear, 2002). Le contexte et l'historique de la conversation doivent être pris en compte lors de la formulation d'un énoncé. De plus, le langage utilisé doit être adapté à l'interlocuteur visé.

Les méthodes les plus simples consistent à utiliser des textes prédéfinis (Weizenbaum, 1966) ou des modèles de messages dans lesquels les valeurs des variables sont complétées. L'approche linguistique, quant à elle, divise le processus de décision selon sur trois tâches à accomplir pour déterminer : les informations à fournir, la façon dont le message est exprimé, et la manière dont on va présenter le message (Reiter, 1994). On peut réunir ces tâches selon les méthodes de planification de documents (la sélection et structuration du contenu), de planification de surface (détail de la phrase et analyse lexicale) et la réalisation de surface (structure de l'énoncé et analyse linguistique) (Rambow *et al.*, 2001; Stone *et al.*, 2003; Stent *et al.*, 2004). Les approches orientées sur les données, en particulier l'apprentissage par renforcement, sont étudiées pour l'application à la gestion du dialogue et la génération de textes à partir de modèles statistiques entraînés sur des ensembles de dialogues (Lemon, 2011; Rieser et Lemon, 2011).

Les méthodes de génération en langage naturel évoluent pour produire des énoncés plus naturels. Bien que les méthodes se reposant sur d'algorithmes d'apprentissages soient de plus en plus étudiées, elles sont encore réservées à des applications de la recherche.

1.2.3 Méthodes de contrôle du dialogue

La gestion du dialogue repose sur le choix d'une méthode de contrôle. Comme pour les tâches de TALN, les méthodes de représentations ont évolué et se divisent selon deux types d'approches (Lison et Meena, 2014) :

- les approches dites *manuelles* : les états du dialogue et la stratégie utilisée pour la sélection de l'action à exécuter sont directement déterminés par des experts et codés dans le système. McTear (2002) distingue trois approches principales : les machines à états, les formulaires et les agents.
- les approches reposant sur des algorithmes d'apprentissage automatique apprennent les transitions à partir d'exemples de dialogues réels.

1.2.3.1 Méthodes par modélisation manuelle

1.2.3.1.1 Machines à états finis

L'une des premières approches étudiées est de modéliser la conversation sous la forme d'une machine à états finis. Ce type d'automate est formellement défini par le quintuple $A = (\Sigma, S, s_0, \delta, F)$. Où, Σ est un ensemble fini (non vide) d'entrées. S est un ensemble fini (non vide) d'états, avec s_0 l'état initial et F l'ensemble des états terminaux. Finalement, une machine à états repose sur des transitions, l'ensemble δ , entre des états dépendants des entrées⁹. Plus particulièrement, cette méthode appliquée à la gestion de dialogue définit chaque élément de la manière suivante. L'ensemble Σ est un ensemble de conditions sur l'énoncé d'entrée (sa représentation formelle et les informations contenues). Une approche simple et intuitive définit l'ensemble des états S par des réponses et (ou) action(s) du système. Par conséquent, les transitions δ sont des fonctions de l'entrée (le résultat du module de compréhension du langage naturel) et de l'état courant du système (Di Fabrizio *et al.*, 2002).

Cette méthode est d'avantage utilisée pour des systèmes orientés vers des tâches précises ou fondées sur des commandes. Dans le premier cas, un état peut représenter une (des) information(s) à recueillir et le résultat produit une action ou une réponse du système (Valenta *et al.*, 2012). Pour le second, une commande a un effet sur le système en fonction d'un état courant (Bourguet, 2003). Cependant, Lcock (2012) applique les FSM à une tâche plus complexe dans un domaines ouvert pour gérer le changement de sujet dans des conversations.

1.2.3.1.2 Formulaires

Dans les approches fondées sur des formulaires (aussi appelées cadres), les systèmes visent à remplir un ensemble d'informations dans des formulaires pré-définis (Goddeau *et al.*, 1996). Cette approche est aussi adaptée aux systèmes orientés vers des tâches pour lesquels un formulaire est une tâche à réaliser et les attributs sont les informations nécessaires pour réaliser

9. Cette définition formelle sera utilisée par la suite, dans notre développement. Nous y ferons références dans les mêmes termes.

cette tâche (Seneff et Polifroni, 2000). De plus, ceci permet d'obtenir des dialogues à initiative mixte. L'utilisateur peut fournir plusieurs informations dans son message, le système identifie un cadre dans l'énoncé, extrait les attributs et demande explicitement à l'utilisateur les informations manquantes pour finir de compléter le cadre (Papineni *et al.*, 1999).

1.2.3.1.3 Agents intelligents

Une conversation peut être définie comme un acte de collaboration et de négociation entre les participants. Cependant, dans les deux approches précédentes, ces notions ne sont pas (ou peu) prises en compte. Les approches basées sur des agents intelligents cherchent à tirer partie de l'aspect collaboratif des systèmes multi agents pour concevoir des systèmes fondés sur des architectures permettant de reproduire des conversations collaboratives (Busemann *et al.*, 2011).

Les approches précédentes se concentrent sur la modélisation du dialogue et sont davantage appliquées à des systèmes orientés vers des tâches. Les approches fondées sur des agents s'appuient aussi sur des systèmes de planification (Allen et Perrault, 1980) pour reconnaître et satisfaire le souhait de l'utilisateur. En intelligence artificielle, un plan est un ensemble d'actions prédéfinies permettant, à partir d'un état initial, d'atteindre un but donné. Chaque action est définie par une (ou des) précondition(s) et a un (ou des) effet(s). Appliquée au contrôle de dialogue, la planification associe des actions à des actes de dialogue. Par conséquent, l'énoncé d'un utilisateur identifie une action, le système infère sur le plan potentiel dont fait partie cette action pour déterminer le but et construire le plan global afin de déterminer la réponse adéquate. La réponse peut amener à donner des informations supplémentaires et non attendue par l'utilisateur car le système aura prédit le but final.

1.2.3.2 Méthodes statistiques

Grâce à l'amélioration des performances techniques et calculatoires des machines, l'intérêt pour les systèmes orientés sur les données a augmenté. Dans de nombreux domaines, les mé-

thodes statistiques et les algorithmes d'apprentissage tendent à être utilisés pour modéliser des comportements. Nous présentons l'application de ces techniques dans le cadre de la gestion de dialogue.

1.2.3.2.1 Apprentissage supervisé

Les méthodes d'apprentissage supervisé utilisent des ensembles de données annotées pour déterminer un modèle des données. Cet ensemble, appelé d'entraînement, est composé de paires (x_n, y_n) , avec $x \in X$ l'ensemble des exemples et $y \in Y$ l'ensemble des annotations. Le modèle déterminé permet de prédire pour une nouvelles entrée x une sortie y .

Dans la cadre des SD, les ensembles de données sont des recueils de conversations. En général, la récolte de données s'effectue sous la forme d'expériences nommées « Magicien d'Oz » dans lesquelles l'utilisateur croit interagir avec une machine alors qu'en réalité c'est un humain qui lui répond. Un utilisateur a tendance à ne pas s'exprimer de la même manière selon qu'il s'adresse à un humain ou une machine. Cependant, la récolte de conversations entre un humain et une machine nécessite d'avoir déjà un système capable de répondre à l'utilisateur. Les expériences « Magicien d'Oz » permettent de pallier au problème.

Appliqués à la gestion du dialogue, les algorithmes d'apprentissage supervisé modélisent le comportement du système. Les exemples sont généralement annotés avec des actes de dialogue (Griol *et al.*, 2008). Cependant, la performance de l'algorithme dépend des données d'entraînement et est sensible à la dimension des données d'entrée et de sortie. En effet, plus le nombre d'actes différents sera important, plus le nombre d'éléments dans l'ensemble d'entraînement devra augmenter pour pouvoir couvrir l'ensemble des possibilités et être représentatif du domaine.

Le développement particulier des *charterbots* qui ne sont pas orientés vers un objectif (Nio *et al.*, 2014) se repose aussi sur une BDD d'exemples. Cependant, la réponse du système est déterminée à partir de mesures de similarités (Banchs et Li, 2012). De même, un large ensemble de données est nécessaire pour être capable de considérer des énoncés divers.

1.2.3.2.2 Apprentissage par renforcement

L'apprentissage par renforcement est une solution pour pallier les différentes limites présentées précédemment et en particulier la difficulté de modéliser une stratégie manuellement. L'objectif de l'apprentissage par renforcement est d'apprendre le comportement générant la récompense la plus importante à long terme. Plus formellement, l'apprentissage par renforcement repose sur cinq notions : une politique (une stratégie), des états, des actions, des récompenses (positives et négatives) et un but. Dans chaque état, une récompense (positive ou négative) est assignée à chaque action possible. L'agent apprend à déterminer une suite d'actions possibles permettant de maximiser le gain total pour atteindre son but (réaliser sa tâche).

Cette méthode est appliquée pour apprendre une stratégie optimale dans le cadre des SD et en particulier ceux visant une initiative mixte. L'objectif du système est de maximiser la satisfaction de l'utilisateur et d'atteindre un but dans un nombre convenable d'échanges. Les premières propositions étaient de modéliser un dialogue sous forme d'un processus de décision Markovien (MDPs) et de déterminer la stratégie optimale à l'aide d'algorithmes d'apprentissage par renforcement (Levin *et al.*, 2000). Dans le cadre des SD vocaux, le processus de décision Markovien partiellement observable (POMDPs) est plus adapté à la prise en compte de l'erreur venant du ASR. L'information partiellement observable est le résultat retourné par le composant de reconnaissance vocale (Young *et al.*, 2013; El Asri *et al.*, 2014b).

Cependant, pour déterminer la politique optimale dans le cadre d'un dialogue l'interaction avec un humain est nécessaire mais fastidieuse à déployer. Les solutions proposées se basent sur des ensembles de données construits sur une méthode de « Magicien d'Oz » (El Asri *et al.*, 2014a).

1.2.3.3 Avantages et Inconvénients

1.2.3.3.1 Méthode manuelles

L'avantage des méthodes fondées sur des FSM repose sur leur facilité de compréhension, de conception et de visualisation de l'espace d'états (Skantze et Al Moubayed, 2012). Cependant,

le nombre de transitions et d'états peut rapidement augmenter et rendre l'implémentation difficile. C'est donc une méthode difficile à appliquer à des dialogues complexes (Ren *et al.*, 2015). Bien que l'approche fondée sur des formulaires donne plus de flexibilité, elle se limite aussi à des utilisations pour des systèmes orientés vers une tâche adaptée à des modélisations en formulaires. En revanche, les méthodes fondées sur les agents tirent parties des méthode d'intelligence artificielle pour les appliquer dans la modélisation de caractéristiques particulières des dialogues (par exemple les initiatives mixtes). Cependant, les transitions entre les états du dialogue ne tiennent pas compte de l'incertitude et nécessitent la compréhension et modélisation préalable du comportement avec des experts (Ren *et al.*, 2015). Ces méthodes font l'hypothèse que le comportement de l'utilisateur peut être parfaitement identifié et cloisonné.

1.2.3.3.2 Méthode statistiques

Les méthodes statistiques visent à déduire des comportement à partir de l'étude de comportements réels et en particulier l'utilisation d'ensembles de données représentatifs de conversation. Bien qu'elle permettent d'obtenir des résultats intéressants sur la génération d'un comportement plus flexible pour l'utilisateur, elles demandent de nombreuses ressources, que ce soit d'un point de vue des données ou des performances de calcul des machines. Les méthodes fondées sur les données sont encore très peu utilisées dans des application commerciales (Pieraccini et Huerta, 2005).

1.3 Système de dialogue et prise de rendez-vous

La prise de rendez-vous est un processus faisant partie de notre quotidien, que ce soit entre individus, avec des institutions ou des entreprises. Cependant, le processus de prise de rendez-vous est souvent réduit à une tâche d'ordonnancement incluant l'optimisation de calendriers et la détermination d'un créneau satisfaisant pour les parties impliquées. L'acte de prise de rendez-vous implique une négociation entre les acteurs et la construction collaborative d'un rendez-vous. Dans le présent projet, l'intégration d'une interface conversationnelle par le texte

entre un système d’ordonnancement et un utilisateur est étudiée pour améliorer un outil de prise de rendez-vous. Nous utiliserons en particulier des échanges par SMS.

Dans cette section, nous présentons des projets utilisant le SMS pour la prise de rendez-vous qui montrent dans quel contexte l’utilisation de ce service peut améliorer le processus. En revanche, ces études n’incluent pas l’automatisation de la conversation et en particulier du traitement du langage naturel dans les textes. Nous présenterons dans un second temps, des systèmes qui étudient la conversation et la prise de rendez-vous en langage naturel.

1.3.1 L’utilisation des SMS pour la prise de rendez-vous

Plusieurs études démontrent les avantages de l’utilisation des SMS dans l’amélioration de la gestion des rendez-vous. Nous avons d’abord identifié des recherches se concentrant sur l’utilisation des SMS en amont de la prise de rendez-vous et en particulier à des fins de rappels. Cependant, dans le cadre de la prise de rendez-vous, les échanges entre le système et le client sont restreints et structurés.

L’utilisation des SMS a été étudiée à des fins de rappels automatisés d’un système vers un client. Le cycle de vie d’un rendez-vous commence lorsqu’une demande est effectuée et se termine lorsque la rencontre a eu lieu. La durée entre ces deux moments peut être plus ou moins longue. Afin de diminuer le risque d’oubli, le recours à des systèmes automatisés de rappels est recommandé¹⁰. L’assiduité est particulièrement étudiée dans le cadre des rendez-vous avec les médecins où les rendez-vous peuvent être fixés longtemps en avance (Downer *et al.*, 2005). Or, que ce soit pour les entreprises ou les services de soins, un rendez-vous où le client (ou patient) ne se présente pas est à la fois une perte de temps et d’argent (Koshy *et al.*, 2008).

L’utilisation des SMS s’est montrée efficace pour l’augmentation de l’assiduité des patients (Downer *et al.*, 2005; Koshy *et al.*, 2008; da Costa *et al.*, 2010). En effet, depuis la mise sur le marché des premiers téléphones mobiles dans les années 1980, l’utilisation des téléphones

10. L’entreprise partenaire, propose actuellement ce type d’outil de rappels par SMS

mobiles ne cesse d'augmenter. En 2014, sept milliards d'abonnements ont été comptabilisés ¹¹. Pour une entreprise, le coût de l'envoi d'un SMS est minime et l'automatisation en est facilement réalisable (Taylor *et al.*, 2012). Plusieurs services sont disponibles pour permettre l'envoi automatique de SMS ¹².

La gestion des files d'attente est une autre tâche de la gestion des rendez-vous propre aux services qui ne permettent pas de prendre des rendez-vous au préalable. Le client se présente et doit s'inscrire sur une liste d'attente en attendant qu'une place se libère ¹³. Afin d'améliorer la satisfaction du client et d'aboutir à un gain de temps pour le service, des systèmes de gestion de file d'attente (Suomi *et al.*, 2007) préviennent le client lorsqu'une place se libère par l'envoi d'un SMS. Le client peut alors accepter ou refuser la place en répondant au message. L'envoi de SMS est un outil efficace pour contacter l'utilisateur et lui permettre de ne pas patienter en salle d'attente, donc d'être joignable facilement et n'importe où.

Enfin WAS-GN (Mohsin *et al.*, 2011) est un système de prise de rendez-vous par internet entre un élève et un professeur. L'étudiant effectue une demande de rendez-vous et le système se charge de valider le rendez-vous avec le professeur. Les auteurs démontrent que l'utilisation du SMS pour notifier une requête sur la plateforme a permis d'augmenter le nombre de réponses. Cependant, l'échange par SMS s'effectue seulement entre le système et le professeur qui peut accepter ou refuser le rendez-vous. Cette approche a été testée en comparant la prise de rendez-vous par l'envoi de courriels et de SMS. Il en résulte que le temps de réponse par le professeur était diminué par l'utilisation du SMS.

L'utilisation des SMS dans le cadre de la prise de rendez-vous semble pertinente pour les institutions et les entreprises. D'abord, comme le service de SMS est intégré aux téléphones mobiles, les utilisateurs sont joignables en tout temps et en tout lieu. De cette manière, on peut diminuer le temps de prise en compte du message. De plus, les SMS sont appréciés pour leur

11. Statistiques issues de journal du Net (2015)

12. Des exemples de plateforme pour le développement d'application pouvant recevoir et envoyer des appels ou SMS : <https://www.twilio.com>, <https://www.plivo.com>, <http://www.octopush.com>

13. L'entreprise partenaire propose un outil de gestion de file d'attente, cependant elle n'utilise pas les appels ou l'envoi de SMS.

faible coût et la possibilité de facilement automatiser l’envoi. Cependant, les études présentées limitent les échanges entre le système et l’utilisateur. Soit les échanges sont unidirectionnels, du système vers l’utilisateur, soit bidirectionnel, mais avec un langage contraint et limité. Dans la section suivante, nous présentons des travaux menés pour le traitement de gestion de la conversation en langage naturel pour la prise de rendez-vous.

1.3.2 La prise de rendez-vous par échanges en langage naturel

Nous avons détaillé différentes méthodes de contrôles de dialogues (*voir* Section 1.2.3). Le choix de la méthode dépend du type de conversation que l’on souhaite entretenir et de la tâche à réaliser par le système. Ici, nous nous intéressons aux systèmes pour la prise de rendez-vous. Par définition le processus de prise de rendez-vous est la détermination d’une date et d’une heure convenant aux personnes impliquées. Busemann *et al.* (2011) s’intéressent aux échanges par courriels et à la construction collaborative du rendez-vous entre plusieurs interlocuteurs. El Asri *et al.* (2014a) s’intéressent, quant à eux, aux échanges vocaux et l’acte de négociation effectué entre deux interlocuteurs.

COSMA (Busemann *et al.*, 2011) est un système multi agents pour la prise de rendez-vous entre plusieurs interlocuteurs, en langage naturel, par courriels. Les auteurs étudient l’automatisation de la collaboration et de la négociation du rendez-vous en tirant partie des systèmes multi-agents. Un secrétaire virtuel est représenté par un agent aidant à la négociation entre les différents interlocuteurs.

NASTIA (El Asri *et al.*, 2014b) est un système de dialogue vocal, pour la gestion de prises de rendez-vous par un client pour l’intervention d’un ingénieur. Les auteurs supposent que la prise de rendez-vous repose sur l’acquisition de trois informations : la date, l’heure et une partie de journée (matin, après-midi et soirée). La conversation est modélisée par une machine à états finis où chaque état représente une phase de la conversation et où un choix parmi un à cinq de types d’actions est possible. L’apprentissage par renforcement est utilisé dans l’automate et repose sur l’ensemble de données DINASTI (El Asri *et al.*, 2014a) conçu pour cette tâche

spécifique afin de déterminer une stratégie optimale. Les auteurs cherchent à évaluer trois types de stratégies de négociation : laisser le client donner ses choix, obtenir les informations par des questions du système ou lister des propositions.

1.4 Conclusion

Cette revue de littérature nous a permis de définir l'étendue des domaines liés aux systèmes de dialogue et leur complexité. Dans le cadre de notre projet, l'utilisation d'une interface conversationnelle était un moyen de réduire les possibilités d'expression pour approfondir la modélisation d'un rendez-vous et de la prise de rendez-vous. En effet, alors que les systèmes de dialogue visent à faciliter les échanges entre un utilisateur et un système, il réduisent la quantité d'informations que le système peut fournir à l'utilisateur. Les études des SD appliquées à la prise de rendez-vous ont confirmé notre hypothèse sur la particularité de ce processus dans un contexte conversationnel mais elles étaient limitées à leur contexte d'étude. Nous nous sommes intéressés au cas particulier des échanges entre un client et une entreprise de services, et à l'intégration d'une interface conversationnelle dans un système existant.

A la suite de ces observations, l'étude a été menée afin de :

- 1 caractériser l'effet du changement du moyen de communication sur la gestion de la prise de rendez-vous ;
- 2 dégager les caractères particuliers d'une conversation pour la prise de rendez-vous.

Les trois chapitres suivants décrivent les travaux réalisés pour répondre à nos objectifs. Le chapitre deux présente l'étude du domaine réalisées à partir d'une récolte de données. Le chapitre trois, présente l'étude d'une machine à états pour le développement du prototype. En effet, la revue de littérature a montré que les méthodes reposant sur les machines à états étaient efficace dans le cas de développement rapide de systèmes de dialogue dirigés vers des buts. Le chapitre quatre étudie la représentation formelle des énoncés sous forme d'intentions et une méthode de contrôle reposant à la fois sur une machine à états et la notion de formulaires.

CHAPITRE 2

PHASE PRÉLIMINAIRE

Nous souhaitons comprendre les aspects particuliers des requêtes envoyées par un client à une entreprise de services pour prendre un rendez-vous. Nous n'avons pas accès à un ensemble de données existant et correspondant au contexte que nous souhaitons étudier. Même si certaines entreprises entretiennent de manière informelle des échanges par SMS avec leurs clients, nous n'avons pas accès à ces messages. En effet, ces échanges n'étant pas réalisés dans une démarche de récolte officielle, aucune méthode d'enregistrement ou de gestion de la confidentialité n'est employée. C'est pourquoi, nous avons décidé de mener notre propre récolte de données.

L'objectif de cette phase préliminaire était d'étudier le comportement de l'utilisateur et du contenu de ses messages dans le cadre d'une prise de rendez-vous par SMS et sans contrainte sur la manière de s'exprimer, afin de :

- identifier les informations fournies par un utilisateur en fonction du service visé avec un intérêt particulier pour les expressions temporelles ;
- dégager une catégorisation des messages pouvant être reçus.

Nous avons réalisé une récolte de données en simulant une prise de rendez-vous entre un client et une entreprise. Pour cela, nous avons mis à disposition un numéro de téléphone représentant l'entreprise fictive, et demandé à des participants d'envoyer des messages.

Dans la première section, nous présentons la méthodologie employée ainsi que les contraintes et limites imposées. Dans la deuxième et troisième section, nous décrivons la phase de récolte, puis l'analyse des messages sélectionnés. Enfin, nous concluons avec un retour sur nos objectifs et notre méthode, ainsi que les axes d'études proposés pour la suite du projet.

2.1 Méthodologie et matériel

Pendant la phase préliminaire nous avons réalisé une récolte de SMS suivie d'une analyse d'un sous ensemble de messages sélectionnés.

La récolte de SMS (*voir* Section 2.2) s'est déroulée en trois étapes. Tout d'abord, nous avons développé une application Web permettant de recevoir des SMS par l'intermédiaire d'un numéro fictif. Puis, nous avons sélectionné un panel de participants. En effet, bien que le temps à disposition était restreint¹, nous souhaitions rencontrer des participants avec des profils hétérogènes touchant à différentes tranches d'âges et à différents milieux sociaux. Enfin, nous avons déterminé des moyens de communication pour mobiliser ces participants et transmettre les instructions. L'outil principal était un site internet présentant le projet et les instructions à suivre pour participer à la récolte.

Nous souhaitions étudier les conversations où un utilisateur amorce une conversation par l'envoi du premier message d'introduction. La récolte s'est donc limitée à ces premiers messages puisqu'ils nous semblaient représentatifs de la requête initiale émise par l'utilisateur en langage naturel.

La phase d'analyse (*voir* Section 2.3) s'est décomposée en trois tâches : le prétraitement des données, l'annotation d'entités et la catégorisation des messages. Nous avons procédé à une annotation de concepts dans les textes, de manière manuelle ou automatique (aidée par un outil d'annotation automatique) afin de dégager une catégorisation des messages.

2.2 Récolte des messages

2.2.1 Application Web

Afin de rendre l'expérimentation plus proche d'une situation réelle, nous avons développé une application Web permettant de recevoir et d'envoyer des messages textes par l'intermédiaire d'un numéro de téléphone. Nous avons utilisé le service Twilio, une « Plateforme en tant que

1. Ce projet s'est déroulé en collaboration avec une entreprise pour une durée de huit mois.

Service » (PaaS)² offrant une suite de services pour le développement d'outils de communication par appels ou par SMS. Le service Twilio communique avec notre application Web déployée sur un serveur par l'intermédiaire de requêtes HTTP. De cette manière, les participants ont pu envoyer des messages à un numéro de téléphone réel. Les messages étaient reçus et stockés par l'application dans une base de données. Les participants étaient libres d'envoyer des messages quand ils le souhaitaient et avec leur propre appareil mobile.

2.2.2 Sélection des participants

L'objectif était de mobiliser un ensemble de participants avec des profils différents parlant couramment le français. Nous avons choisi trois groupes (*voir* Tableau 2.1) pour former un panel de personnes âgées de 20 à 45 ans. D'abord, dans le cadre d'un cours de forage de données³, nous avons présenté le projet aux étudiants et aux professeurs en génie des technologies de l'information. Puis, par l'intermédiaire des réseaux sociaux, nous avons contacté des étudiants et jeunes travailleurs âgés de 20 à 30 ans. Enfin, en contactant les employés de l'entreprise partenaire connaissant le processus de prise de rendez-vous et les besoins des clients.

Tableau 2.1 Détail des profils des participants sélectionnés.

Groupe	Cours	Réseaux sociaux	Entreprise (Agendize)
Âge	22-50	20-30	25-50
Profession	Étudiants Professeurs en TI	Étudiants Jeunes travailleurs	Employés de l'entreprise
Domaine	Technologie de l'Information (TI)	divers	divers
Intérêt pour le projet	fort	moyen	fort
Utilisation des SMS	Quotidien	Quotidien	Quotidien
Moyen de communication	Site internet	Réseaux sociaux Site internet	Courriel Site internet

2. Traduction de l'anglais « Platform As A Service »

3. Un cours enseigné par Madame Sylvie Ratté : MTI830

2.2.3 Choix des moyens de communication

Le site internet (voir Figure 2.1) était notre outil principal de communication, regroupant à la fois la description du projet, les instructions à suivre et le numéro de téléphone à utiliser. La mobilisation des participants s'est effectuée soit par l'envoi de courriels, soit par l'utilisation d'un réseau social. Dans les instructions, nous avons présenté des exemples de contextes de prises de rendez-vous limités aux trois types de services choisis (coiffeurs, restaurants ou services médicaux).

Participer

Pour participer : rien de plus simple.
Il vous suffit de lire une des situations présentées ci-dessous et d'envoyer un sms au numéro indiqué, comme si vous preniez rendez-vous "pour de vrai".



Coiffeur

Essayez de vous mettre dans une situation où vous devez prendre rendez-vous avec un coiffeur. Vous pouvez vous adresser à une personne en particulier ou un salon en général.

Participer



Restauration

Un restaurant propose de prendre rendez-vous pas SMS en mettant à disposition un numéro de téléphone.

Participer



Les Soins Médicaux

La prise de rendez-vous avec un médecin peut-être différente en fonction de l'importance du rendez-vous.

Participer

A propos

Cette collecte de données s'inscrit dans un projet de recherche en maîtrise, en Technologie de l'Information, en collaboration entre l'entreprise Montréalaise Agendize Services et le Laboratoire LINCIS de l'école de Technologie Supérieure de Montréal.

La prise de rendez-vous en ligne est un produit déjà largement proposé, mais qui ne réponds pas toujours aux besoins des utilisateurs. Le but du projet est alors d'étudier la notion de prise de rendez-vous par sms et en particulier la conversation qui s'effectue entre deux personnes.

Mais ceci implique une étude des sms échangés et par conséquent une récolte de données afin de réaliser un corpus.




Suivez Nous



Figure 2.1 Page de présentation du projet

À la suite d'une première période de récolte, nous avons dû modifier et préciser les instructions. Le contexte a été clarifié et des exemples ont été proposés pour guider les participants. Bien

que la situation de prise de rendez-vous semble commune, les utilisateurs ont exprimé le besoin d’être guidés.

Nous avons mis à jour les instructions et proposé aux utilisateurs de préciser leur requête :

- Vous n’êtes pas pressé, mais vous désirez prendre un rendez-vous pour une coupe. Comment pourriez-vous exprimer votre demande ?
- Vous avez un impératif (un événement exceptionnel, un rendez-vous d’affaires, etc.). Comment exprimeriez-vous la notion d’urgence ?
- Vous souhaitez prendre un rendez-vous avec un coiffeur pour une coupe de cheveux. Vous ne connaissez pas le coiffeur (ou salon), mais un(e) ami(e) vous l’a fortement recommandé et vous a donné son numéro de téléphone mobile.

Initialement le langage des utilisateurs n’était pas limité. Cependant, dans la seconde phase de récolte nous avons précisé qu’ils devaient imaginer s’adresser à un système reproduisant le comportement humain.

Finalement, à la suite de cette période de récolte nous avons obtenu un ensemble de SMS destinés à la prise de rendez-vous avec trois types d’entreprises.

2.3 Analyse

2.3.1 Prétraitement des messages

Plusieurs tâches de prétraitement ont été effectuées sur l’ensemble des messages reçus :

- suppression des messages considérés comme hors sujet ;
- classification par types de services ;
- anonymisation et suppression des références à des noms propres.

2.3.2 Annotation d'entités

Par la suite, nous avons parcouru l'ensemble des messages sélectionnés pour distinguer les concepts qui permettent d'en qualifier les caractéristiques. Nous avons annoté manuellement chaque message et regroupé les termes ou expressions en familles de concepts classés en fonction des types de services proposés (voir Tableau 2.2).

Tableau 2.2 Concept associé aux informations relevées

Concept	Coiffure	Médecin	Restaurant
Quoi	Service	Raison (symptômes)	nombre de personnes
Qui	Nom propre (employé)	Nom propre (Médecin)	
Quand	Expression temporelle (ET)	ET	ET
Autre	Urgence	Urgence	Information complémentaires

Bien que les concepts soient différents, une structure commune est identifiable. On peut synthétiser l'information sous la forme de questions : *quoi*, *qui* et *quand*. Le *quoi*, l'objet de la requête peut être le nom d'un service clairement identifié (« une coupe », « un détartrage », « une table ») ou implicitement exprimé sous forme d'un besoin (« j'ai besoin de rafraîchir ma coupe », « j'ai mal au ventre depuis deux jours », « organiser un repas d'anniversaire »). La précision du nom de l'employé demandé (*qui*) a été relevée et semble pertinente que pour le cas des coiffeurs et des médecins. Enfin, les expressions temporelles ont été relevées (*quand*).

Dans un processus de prise de rendez-vous, la notion de temps est un élément indispensable. En effet, un rendez-vous est par définition la rencontre d'une ou plusieurs personne(s) à une date et une heure précises. L'acte de prise de rendez-vous vise à déterminer cette date et heure convenant aux parties impliquées. Cependant, on remarque que les expressions temporelles ne sont pas les seuls éléments identifiés et qu'elles ne sont pas présentes dans tous les messages.

La section suivante détaille l’analyse plus approfondie des expressions temporelles à l’aide d’un outil d’annotation et de normalisation automatique.

2.3.3 Annotation des expressions temporelles

2.3.3.1 Présentation de l’outil HeidelTime

HeidelTime (Strötgen et Gertz, 2010) est un annotateur d’expressions temporelles, inter-domaine et multi-langue, développé par l’université d’Heidelberg. Les expressions temporelles dans les textes sont annotées selon les recommandations TIMEX3 en utilisant le langage à balise TimeML (Letzte Änderung, 2015). La normalisation de chaque expression annotée est effectuée en fonction du domaine et de la langue du texte, et repose sur une méthodologie à base de règles. Chaque langue est définie selon des ressources décomposées en trois fichiers : les patrons, les informations sur la normalisation et les règles de normalisation.

Grâce à la définition de ressources adaptées à la langue française, l’outil peut être utilisé sur des textes en français. L’annotateur a été évalué sur le corpus French TimeBank⁴. Les résultats sont comparables à ceux obtenus à partir de la version anglophone sur des corpus en anglais. Les ressources françaises ont été construites, d’une part, à partir de la traduction des ressources anglaises et espagnoles. D’autre part, des règles spécifiques applicables au français ont été créées prenant en compte les règles grammaticales du français.

Pour la langue anglaise, quatre types de textes sont proposés : narratifs, journalistiques, langue courante (SMS, Tweets, etc.) ou articles scientifiques. Dans le cadre des SMS, le corpus Time4SMS (Strötgen et Gertz, 2012) a été utilisé, mais est seulement disponible pour l’anglais. Par conséquent, les ressources françaises permettent d’annoter seulement les deux premiers types de textes.

4. French TimeBank (Bittar *et al.*, 2011) est un corpus de référence contenant des textes journalistiques dans lesquels les expressions temporelles et les événements sont annotés en respectant la norme ISO-TimeML. Sa création repose sur les travaux déjà effectués en anglais (TimeBank (Pustejovsky *et al.*, 2003)) et a impliqué une modification de l’ISO-TimeML (Pustejovsky *et al.*, 2010) qui était à l’origine destiné à l’anglais.

Les données sont préalablement annotées avec TreeTagger (Schmid, 1995), un outil de référence en terme d'étiquetage grammatical et applicable au français. HeidelTime utilise le corpus French TimeBank (Bittar *et al.*, 2011) et les ressources adaptées pour l'étiquetage des expressions temporelles en français (Moriceau et Tannier, 2013).

2.3.3.2 Annotation des données récoltées

Afin d'analyser les expressions temporelles utilisées dans les messages, nous avons utilisé l'annotateur de référence HeidelTime. L'objectif était de pouvoir identifier les types d'expressions temporelles utilisées dans les textes des messages récoltés à partir du résultat de l'annotation effectuée par HeidelTime selon la norme ISO TimeML. Cet outil devant être utilisé dans le prototype, nous avons d'abord évalué l'annotation appliquée à nos données brutes. Puis nous avons déterminé des classes de messages en fonction des expressions temporelles. En revanche, nous avons relevé des erreurs d'étiquetage et de normalisation.

Les causes d'erreurs d'étiquetage relevées étaient :

- les expressions temporelles avec des erreurs d'orthographe ;
- les chiffres considérés comme une date ou une durée ;
- les expressions temporelles considérées comme deux expressions distinctes (par exemple, pour le cas de « vendredi prochain matin », l'algorithme a étiqueté d'une part « vendredi », et d'autre part, « prochain matin » comme deux expressions distinctes).

Nous avons relevé deux causes d'erreurs de normalisation. D'abord, une règle⁵ normalise un jour avec une date dans le passé⁶. Puis, lorsque le lien entre une date et une heure n'est pas reconnu, en particulier dans le cas où l'heure apparaît avant la date, ceci conduit à une erreur dans la date normalisée.

5. Nous avons dû utiliser le type journalistique proposé par HeidelTime.

6. Pour la phrase « je voudrais un rendez-vous jeudi. », la date normalisée de « jeudi » correspond au jeudi précédent dans le calendrier.

Finalement, le résultat de l’annotation et de la normalisation des expressions temporelles nous a permis de distinguer quatre situations⁷ :

- une date et une heure précises sont proposées par l’utilisateur ce qui permet de définir une demande de rendez-vous complète : « *J’aimerais me faire couper les cheveux **le 24 août** prochain à **13h30*** » ;
- une date et (ou) une heure sont identifiée(s), mais pas assez précise(s) pour définir un rendez-vous : « *une teinture **lundi** en **soirée*** ». Dans cet exemple, le terme « soirée » est identifié comme une partie de journée ;
- des expressions temporelles ont été relevées, mais les informations disponibles ne suffisent pas à leur donner un sens : « *bonjour Dre Séguin j aimerais faire un detartrage global avant le **printemps*** ». Le terme printemps est identifié comme une date particulière (2016-SP) ;
- aucune expression temporelle n’est identifiée : « ***URGENT!!!** J’ai une terrible rage de dents !* ». L’information fournie n’est pas suffisante pour déterminer un RV.

Dans ces différentes situations, nous remarquons que seuls les deux premières permettent d’effectuer directement une requête pour déterminer un rendez-vous. En revanche, les deux dernières impliquent une analyse sémantique plus approfondie pour donner un sens à l’expression temporelle.

2.4 Résultats

Les objectifs de cette récolte étaient, d’une part, d’observer le comportement des utilisateurs face à un système de prise de rendez-vous par SMS, et d’autre part, d’analyser le contenu de leurs messages. Nous présentons dans cette section, les observations effectuées durant la phase de récolte, puis la catégorisation des messages obtenus à la suite de l’analyse.

7. Les exemples cités sont tirés de l’ensemble de messages récolté.

2.4.1 Phase de récolte des messages

Nous avons mené cette expérience avec 37 participants volontaires (*voir* tableau 2.2) et déjà familiers de l'utilisation des téléphones mobiles et des nouvelles technologies. Nous avons fait l'hypothèse qu'ils seraient à l'aise avec l'idée d'utiliser des messages textes pour interagir avec un système. L'un des objectifs de l'utilisation des interfaces conversationnelles est de pallier l'inconvénient d'une interface graphique qui est de devoir comprendre le fonctionnement du système au préalable. En utilisant un échange en langage naturel on cherche à rendre l'interaction plus naturelle et à faciliter la commande du système. Cependant, l'expérience réalisée a montré que les utilisateurs étaient en demande d'informations supplémentaires sur le fonctionnement du système et, par conséquent, du langage qu'il devait employer dans leurs messages.

Dans notre expérience, l'utilisateur est à l'initiative de la conversation. Étant donné qu'il envoie le premier message amorçant la conversation, il n'est pas guidé par un message d'introduction envoyé par le système. La première phase de la récolte et les échanges entretenus avec les participants ont démontré la nécessité de guider l'utilisateur. Par exemple, les messages de type « j'ai besoin d'un rendez-vous » montrent que l'utilisateur souhaite initier un contact avec l'entreprise, mais qu'il est en attente de questions pour être guidé quant aux informations à fournir. On retrouve des situations comparables avec l'utilisation d'interface graphique où l'utilisateur est guidé par les éléments graphiques et les instructions affichées.

Dans les messages reçus, soit les personnes étaient très ouvertes et s'exprimaient comme si elle ne s'adressait pas à une machine, soit elles étaient très directives. Dans une seconde phase de récolte, nous avons demandé aux participants de se placer dans une situation où ils dialoguaient avec un système reproduisant le comportement d'un humain. En effet, l'utilisateur ne s'exprime pas de la même manière lorsqu'il sait qu'il communique avec une machine ou avec un humain. Mais à l'origine, nous avons fait le choix de ne pas limiter les participants pour avoir une image plus similaire d'une situation réelle. De cette manière, nous avons pu observer différents types d'échanges.

Finalement, après deux mois de mobilisation et de récolte, nous avons retenu 112 messages provenant de 37 participants. Près de la moitié des messages reçus étaient destinés à des coiffeurs alors que l'autre part était également répartie entre les restaurants et les médecins.

2.4.2 Catégorisation des messages

A partir de la définition de la prise de rendez-vous suivante : la détermination d'une date et d'une heure pour une rencontre, nous avons porté un intérêt particulier aux expressions temporelles dans les messages. Mais nous avons aussi relevé une liste de concepts relatifs à la prise de rendez-vous. Cependant, nous avons observé que les messages n'étaient pas exclusivement des requêtes de rendez-vous. Le client peut demander des informations particulières sur les services ou directement des disponibilités.

La catégorisation présentée dans la figure 2.2 résume les résultats de cette phase de récolte et d'analyse des messages. L'ensemble des messages a été divisé en deux classes : les *requêtes* et les *demandes*. Les *demandes* concernent des informations particulières sur l'entreprise. La *requête* est l'acte de vouloir prendre un rendez-vous en fournissant plus ou moins d'informations (par exemple : la raison, le service, l'employé, etc.).

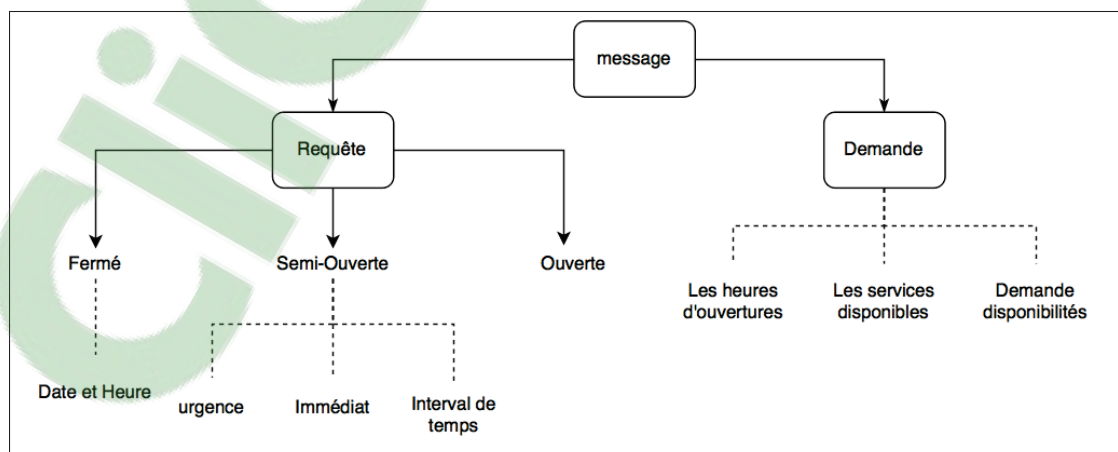


Figure 2.2 Catégorisation des messages

A partir de l'analyse des expressions temporelles nous avons distingué plusieurs types de requêtes :

- fermée, si une date et une heure précises sont données : « Prise de rendez-vous **Lundi 22 juin à 9h** pour nettoyage dentaire. » ;
- semi ouverte, une expression temporelle peu précise : « coupe pour garçon jeudi **en soirée** » ou une expression temporelle n'est pas nécessairement identifiée mais une notion d'urgence est exprimée « URGENT ! [...] » ;
- ouverte, aucune expression temporelle : « je veux me faire couper les cheveux ».

2.5 Conclusion

2.5.1 Retour sur les objectifs

Pour réaliser la récolte, nous avons mis à disposition un numéro de téléphone fictif et mobilisé des participants. Sur une campagne de deux mois, nous avons recueilli une centaine de messages qui nous ont permis de répondre aux questions définies au préalable et de dégager une première catégorisation des messages entrants dans le cadre de la prise de rendez-vous.

Notre hypothèse initiale était que la construction d'une requête pour prendre un rendez-vous pourrait être effectuée à partir de l'identification des expressions temporelles dans les messages. Cependant, toutes les expressions identifiées ne permettent pas de construire une requête complète. De plus, d'autres éléments pertinents ont été identifiés dans les messages. Nous avons aussi identifié des types de messages que nous n'avions pas considéré dans le contexte énoncé. Finalement, nous avons présenté le résultat de ces observations par une catégorisation des messages selon deux classes principales : *demande* et *requête*.

Nous avons recueilli des retours auprès des participants quant à l'utilisation potentielle de ce type de service. Leurs réticences et leurs questions ont montré la nécessité d'être guidés. Ils ont insisté sur l'importance de pouvoir atteindre leur objectif dans un nombre d'échanges acceptables.

2.5.2 Limites de la méthodologie

La méthode que nous avons employée se limitait à la récolte des messages provenant d'un client vers une entreprise.

Nous avons relevé des limites auxquelles nous proposons les solutions suivantes :

1. Le recueil de messages uniques et unidirectionnels : la mise à disposition d'une ressource pour répondre aux messages nous aurait permis, d'une part, de relever un nombre plus important de messages et d'autre part, de motiver les participants.
2. Le choix des instructions : nous n'avons pas été clairs sur nos premières intentions quant à la récolte. Notre ambition initiale était de récolter des messages s'adressant à des entreprises réelles. Par la suite, nous nous sommes rendu compte de la difficulté de traitement et avons modifié nos instructions en cours de campagne pour que les participants s'adressent davantage à un système.

En ne considérant que les messages d'introduction, nous avons observé que les messages composés d'une date et d'une heure précises n'étaient pas majoritaire. Le second élément essentiel relevé est le service demandé ou la raison pour laquelle on souhaite réserver un rendez-vous. Dans le cadre des médecins, l'identification d'un service se résume plutôt à l'expression d'une raison pour laquelle il y aurait un besoin.

Notre étude s'est appuyée sur le développement d'un prototype de système de dialogue par SMS intégrant les fonctionnalités du système d'ordonnancement actuel. En s'appuyant sur les observations réalisées pendant cette phase de compréhension des données, nous avons proposé deux approches de développement pour répondre à nos objectifs initiaux que nous présentons dans les deux chapitres suivants.

CHAPITRE 3

MACHINE À ÉTATS

L'interface graphique de l'outil de prise de rendez-vous en ligne utilisé permet à l'utilisateur de compléter les informations requises afin que le système puisse proposer une liste de créneaux disponibles. L'utilisateur est guidé par l'interface pour remplir les informations les unes à la suite des autres, jusqu'au choix d'une date et d'une heure dans un calendrier. Finalement, l'utilisateur valide son choix pour confirmer le rendez-vous.

L'objectif du prototype était de représenter un processus similaire (voir Figure 3.1) en utilisant une interface conversationnelle. Dans le premier chapitre (voir Chapitre 1 : cadre théorique), nous avons présenté différentes approches de contrôle du dialogue. Dans le cadre de ce prototype, nous avons étudié l'approche basée sur une machine à états similaire au processus actuel.

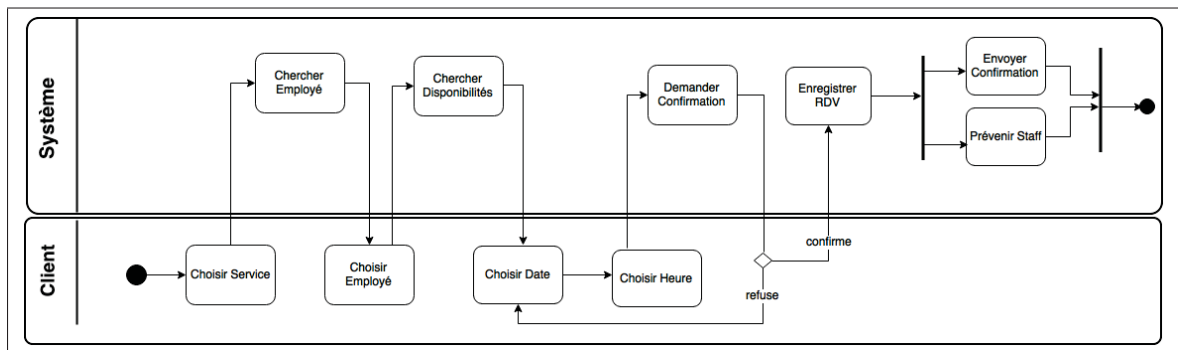


Figure 3.1 Diagramme d'activité : processus de prise de rendez-vous

La première section présente nos hypothèses et nos définitions des concepts utilisés. **La seconde section** présente la méthodologie employée pour le développement du prototype, ainsi que les outils utilisés. **La troisième et la quatrième section** développent respectivement la modélisation et l'implémentation du prototype. **La dernière** conclut par une démonstration de nos résultats et un retour sur nos objectifs.

3.1 Hypothèses et Définitions

3.1.1 Requête de l'utilisateur et rendez-vous

Pendant la phase préliminaire, en étudiant les expressions temporelles, nous avons montré que l'utilisateur n'effectuait pas explicitement une demande pouvant être considérée par le système comme une requête complète (contenant toutes les informations requises).

Cette observation nous a amenée à revoir la définition d'un rendez-vous. Dès que l'utilisateur prend contact avec une entreprise, il émet le souhait de prendre un rendez-vous, même si sa requête est incomplète. Le rendez-vous est l'objet qui se construit au fil des échanges pendant lesquels les valeurs de ses attributs sont identifiées. Finalement, un rendez-vous est considéré comme complet lorsque tous les éléments requis ont été complétés et qu'il a été confirmé par le client et le système (représentant l'entreprise de services).

3.1.2 La conversation à modéliser

L'objectif est de modéliser une conversation dont le but est de déterminer avec le système un rendez-vous. Dans la phase préliminaire (*voir* la Section 2.4.2), nous avons identifié deux principaux types de messages. Cependant, dans le cadre de ce prototype nous sommes limités aux messages de type *requête*.

L'objectif de la conversation est de déterminer une valeur pour chaque attribut du rendez-vous qui conviendrait à la fois au client et à l'entreprise (représentée par le système d'ordonnancement). Le client est à l'initiative, son premier message amorce la conversation. En revanche, c'est le système qui guide la suite des échanges. Elle s'achève lorsqu'un rendez-vous est convenu et qu'il est enregistré dans le système d'ordonnancement.

3.2 Méthodologie et matériel

3.2.1 Méthodologie

Pour entretenir une conversation avec un humain, un SD doit réaliser les tâches suivantes¹ : la compréhension du langage naturel, la gestion du dialogue et la génération d'une réponse en langage naturel. Nous avons présenté plusieurs approches pour la réalisation de chacune de ces tâches. En considérant nos objectifs (présentés en introduction), les hypothèses fixées (voir section 3.1), et nos contraintes de temps² nous avons sélectionné les méthodes suivantes :

- **Contrôle du dialogue :**

Dans la littérature (voir chapitre 1 : *cadre théorique*), nous avons présenté différentes méthodes de contrôle du dialogue divisées selon les approches : manuelle et statistique. En considérant la conversation à modéliser³, nous avons fait le choix d'une méthode fondée sur une machine à états. Nous avons modélisé le dialogue par une suite déterminée d'échanges entre l'utilisateur et le système fondée sur le processus présenté dans la figure 3.1.

- **Compréhension du langage naturel :**

La compréhension du langage naturel se limite à l'extraction des informations suivantes dans les textes : une date, une heure et un service. La date et l'heure sont extraites à l'aide de l'outil HeidelTime⁴. La reconnaissance des expressions temporelles dans les textes est améliorée grâce à l'utilisation d'expressions régulières et à la définition de règles. Le service est extrait à l'aide d'un dictionnaire de mots clés.

- **Génération de la réponse :**

La réponse est conçue par le système en langage naturel à partir d'un ensemble de modèles de messages prédéterminés et composés de variables pour lesquelles les valeurs de ces dernières sont fixées lors de la génération.

1. Cette définition est détaillée dans le Chapitre 1 : *Cadre théorique*

2. Le projet s'est déroulé pendant 8 mois avec l'entreprise partenaire

3. Nos définitions et hypothèses sont présentées dans la section 3.1.2

4. Nous avons présenté cet outil dans le Chapitre 2

Dans la continuité des travaux réalisés dans la phase préliminaire, nous avons implémenté une application communiquant avec le service Twilio, pour la réception de messages textes, et le service Agendize⁵, pour la gestion de l'ordonnancement. Nous avons effectué une phase de conception qui a permis de modéliser les données de l'application, la machine à états et les fonctions du système. Les outils utilisés pour l'implémentation sont détaillés dans la section 3.2.2.

3.2.2 Choix technologiques

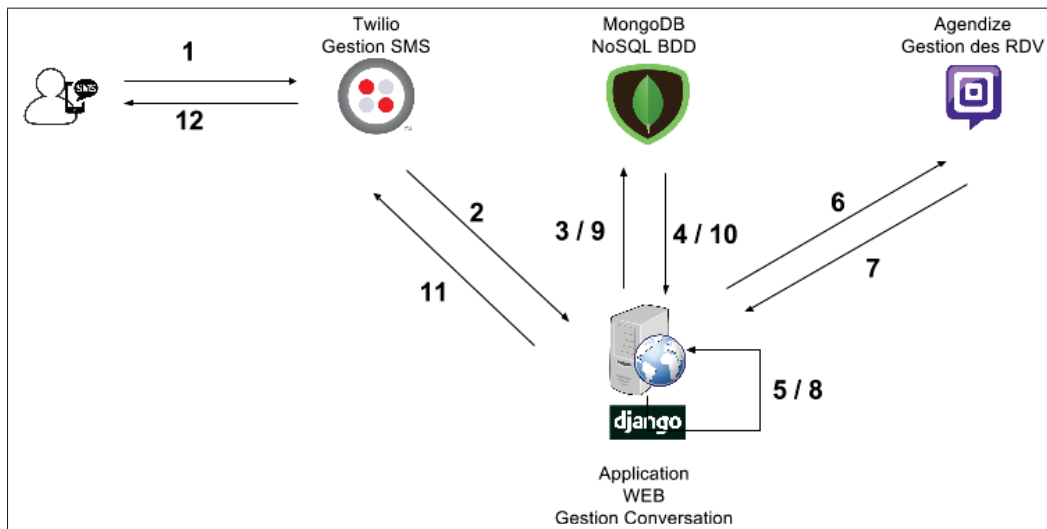


Figure 3.2 Architecture de l'application

La figure 3.2 présente l'architecture générale de l'application, les différents services utilisés et le flux numéroté des échanges réalisés entre les services :

- **Framework Django :**

Nous avons développé une application Web, basée sur le framework Django, utilisant le langage python. Ce framework suit une architecture particulière : modèle, vue, template (gabarit) (MVT). Le modèle décrit les données du système et permet d'accéder à la base donnée. Une vue est le point d'entrée de l'application Web. C'est l'élément comparable au

5. Service d'ordonnancement actuellement proposé par l'entreprise partenaire.

contrôleur dans une architecture Modèle Vue Contrôleur (MVC). La vue reçoit des requêtes HTTP et retourne une réponse. Elle est liée à une fonction de l'application et, dans certains cas, à un gabarit. Un gabarit est la présentation d'une page Web au format HTML. Les valeurs des données présentées dans un gabarit sont déterminées par la vue.

- **Python, packages et modules :**

Nous avons choisi d'utiliser le langage de programmation python. Un fichier « .py » réunit un ensemble d'instructions exécutables pouvant être regroupées en fonctions. Afin d'organiser les fonctions d'un programme, il est possible de créer des *packages* composés de modules. Un package est un dossier contenant un ensemble de fichiers python appelés module.

- **Base de données NoSql :**

Une base de données (BDD) NoSQL permet de stocker des données sous forme de documents contenant des paires attribut-valeur. Contrairement aux BDD SQL classiques, elles ne nécessitent pas la définition d'un schéma au préalable. Dans le cas expérimental qui nous intéresse ici, nous pouvions être amenés à changer notre modèle de données, c'est pourquoi, pour stocker les données de l'application, nous avons opté pour l'utilisation de MongoDB.

- **Service Twilio :**

Twilio est une « Plateforme en tant que Service » (PaaS) offrant une suite de services pour le développement d'outils de communication par appels ou par SMS. Une librairie est disponible pour l'utilisation du service Twilio à partir d'une application Django. Il est nécessaire de créer une vue permettant de recevoir des requêtes provenant du service Twilio. La requête contient un objet de type TWIML contenant les informations sur le message et l'expéditeur.

- **API de l'entreprise partenaire :**

L'entreprise partenaire du projet, développe une suite logicielle pour accompagner les entreprises dans la gestion de leur clientèle. Elle dispose, entre autres, d'un système de prise de rendez-vous. Une entreprise peut définir un ensemble d'éléments pour la prise de rendez-vous, par exemple : les heures d'ouverture, les types de services proposés et leur durée, les

employés et leurs disponibilités. L'API permet d'accéder et d'interroger le système pour gérer les informations sur plusieurs ressources : l'entreprise, les clients, les employés, les rendez-vous et les services proposés. Pour chacune des ressources, une liste de fonctions est proposée. Par exemple pour les rendez-vous on peut : ajouter, supprimer, mettre à jour et lister des rendez-vous disponibles. C'est avec cette API que l'application communique pour gérer la prise de rendez-vous.

3.3 Modélisation

3.3.1 Modélisation de la machine à états

Les hypothèses (voir Section 3.1) ont permis de modéliser la machine à états. Les « états » représentent les phases de l'évolution d'un rendez-vous au cours de la conversation. Dans un premier temps, en fonction des données contenues dans le premier message (il peut contenir plus d'une information), le système demande les informations manquantes à l'utilisateur pour formuler une requête auprès de l'ordonnanceur. Lorsque les informations sont complétées, le système effectue une requête à l'ordonnanceur et retourne, en langage naturel, le résultat de cette requête.

Une machine à états est décrite de manière formelle par le quintuple $A = (\Sigma, S, s_0, \delta, F)$ (voir Section 1.2.3.1.1). Par conséquent, nous avons défini l'ensemble des états S suivant : *initial*, *attente service*, *attente date*, *confirmé*, *refusé* et *enregistré*. L'état s_0 est *initial* et l'unique état final de F est *enregistré*. La fonction de transition est manuellement implémentée à partir de règles dépendantes, à la fois des informations contenues dans le message (date, heure et service) et du résultat de la requête effectuée auprès de l'ordonnanceur. Les transitions entre les états sont présentées dans la Figure 3.3 et décrites par la suite.

Étant donné que l'utilisateur initie la conversation, lors de la réception du premier message, le système est dans l'état *Initial*. Le système reste dans cet état tant que le message reçu ne contient pas au moins une des informations recherchées. Les états *Attente date* et *Attente ser-*

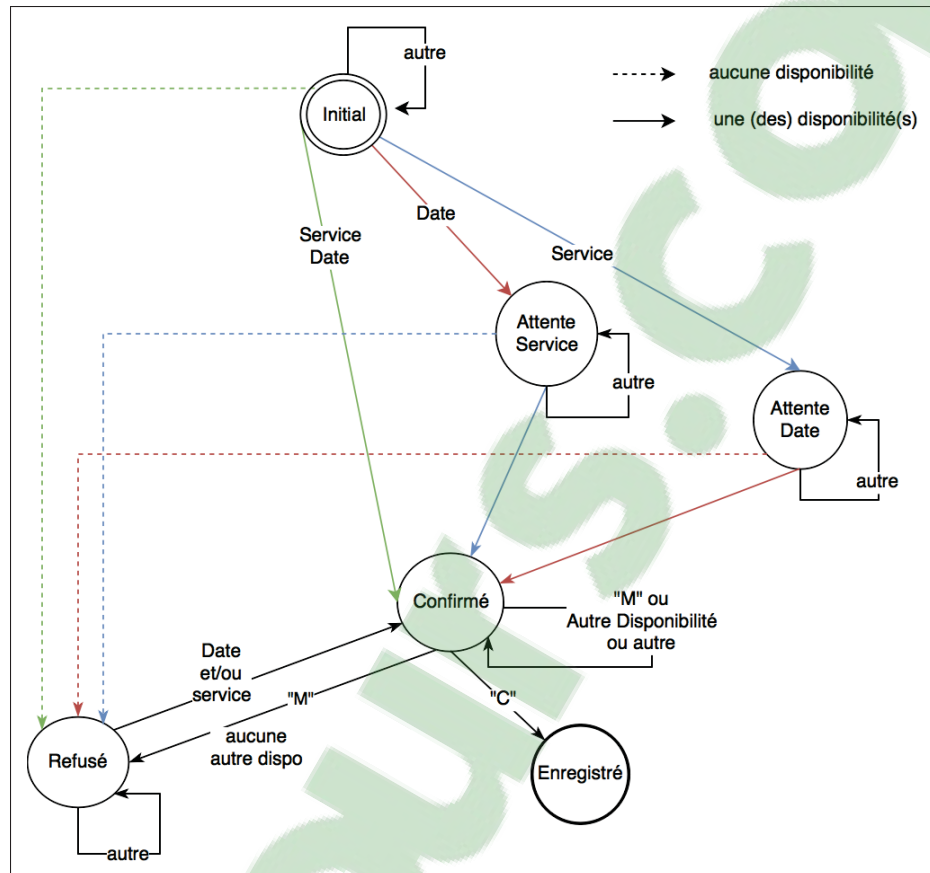


Figure 3.3 Machine à états pour le contrôle de la conversation

vice dépendent des informations contenues dans le message et dans la conversation. Dans l'état *Confirmé*, le système est en attente de la confirmation des informations par l'utilisateur. Si la réponse est positive alors l'état *Enregistré* est atteint et la conversation est terminée. En revanche, si la réponse est négative, soit le système peut formuler une nouvelle proposition et il reste dans l'état courant, soit il demande de nouvelles informations pour modifier la requête et il passe dans l'état *Refusé*. L'état *Refusé* est un état intermédiaire dans la construction du rendez-vous. Dans cet état, toutes les informations requises ont préalablement été collectées, mais n'ont pas été validées par le client ou bien aucun créneau disponible ne correspondait aux critères. Si une nouvelle information est fournie, le système passe dans l'état *Confirmé*. Pour tout autre message, le système reste dans l'état *Refusé*. Finalement, l'état final *Enregistré* est atteint lorsque le rendez-vous est confirmé par le client et le système.

Dans chaque état, une boucle *autre* a été ajoutée pour traiter le cas des messages qui ne sont pas compris ou qui ne correspondent pas aux attentes du système. Dans ce cas, le système retourne un message et reste dans l'état courant.

3.3.2 Modélisation des données

À partir des hypothèses et des définitions (voir section 3.1), nous avons modélisé les données (voir Figure 3.4) pour l'élaboration des documents de notre base de données NoSQL. L'élément principal est le client identifié par son numéro de client dans le système Agendize et son numéro de téléphone. La conversation active est celle qui est en cours entre le système et le client. L'historique représente la liste de toutes les conversations antérieures.

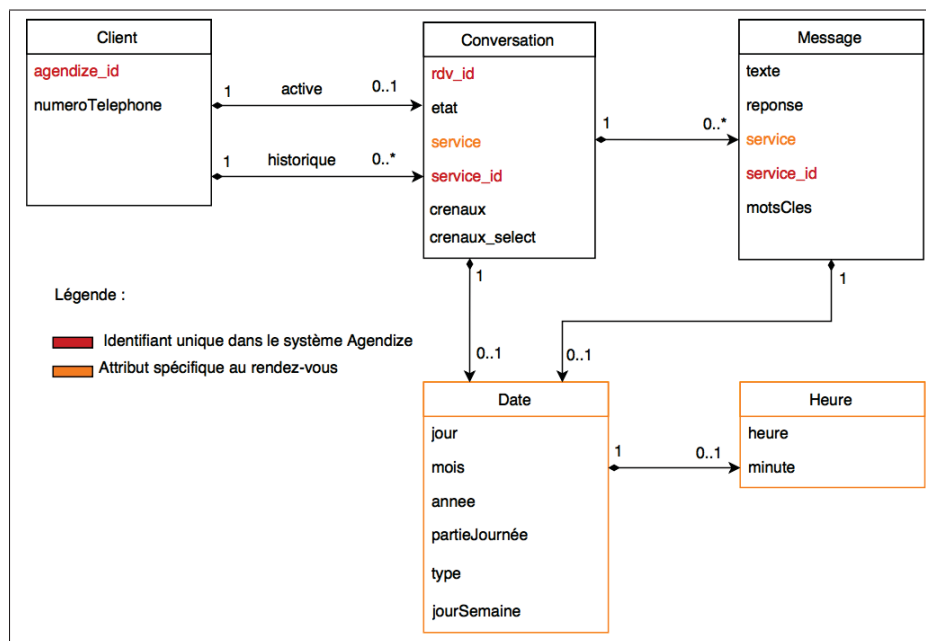


Figure 3.4 Modèle de données

Une conversation contient une liste de messages et les informations concernant un rendez-vous (en orange). Une conversation est la modélisation de l'ensemble des échanges effectués entre un client et un système autour de la construction d'un rendez-vous. Par conséquent, elle ne contient qu'un seul rendez-vous. Les attributs du rendez-vous (en orange) sont donc confondus

avec les attributs de la conversation. Les attributs *créneaux* et *créneaux_select* représentent, respectivement, la liste des créneaux disponibles par rapport à une requête (s’il y a lieu) et la liste des créneaux déjà proposés au cours de la conversation. L’attribut *etat* est l’état courant de la conversation dans la machine à états (voir Section 3.3.1).

Le texte d’un message peut contenir une date, un service ou un mot clé. Compte tenu de la limitation d’une partie de la conversation, l’attribut *motCles* représente le mot clé extrait pour le cas de la confirmation (« M » ou « C »).

Enfin, une date et une heure sont décrites par plusieurs attributs correspondant à la valeur normalisée à la suite de l’extraction dans un texte. L’élément type permet la distinction entre :

- une date précise : une date et une heure : *_ONE_*;
- intervalle (entre deux heures) : *_BETWEEN_*;
- choix entre deux heures : *_OR_*.

3.3.3 Décomposition en modules

À partir de la définition des systèmes de dialogue (voir Chapitre 1) nous avons décomposé notre système en cinq modules (voir Figure 3.5) permettant de couvrir les fonctions de :

- gestion des échanges dans la conversation (rouge) ;
- traitement d’un message en langage naturel (orange) ;
- gestion des rendez-vous (violet) ;

Les description du rôle de chaque module est détaillée dans le tableau 3.1.

3.4 Implémentation

Pour rendre le système opérationnel, nous avons d’abord créé une entreprise de coiffure fictive dans le système d’ordonnancement. Nous avons déterminé un ensemble fini de services pouvant être fournis par l’entreprise.

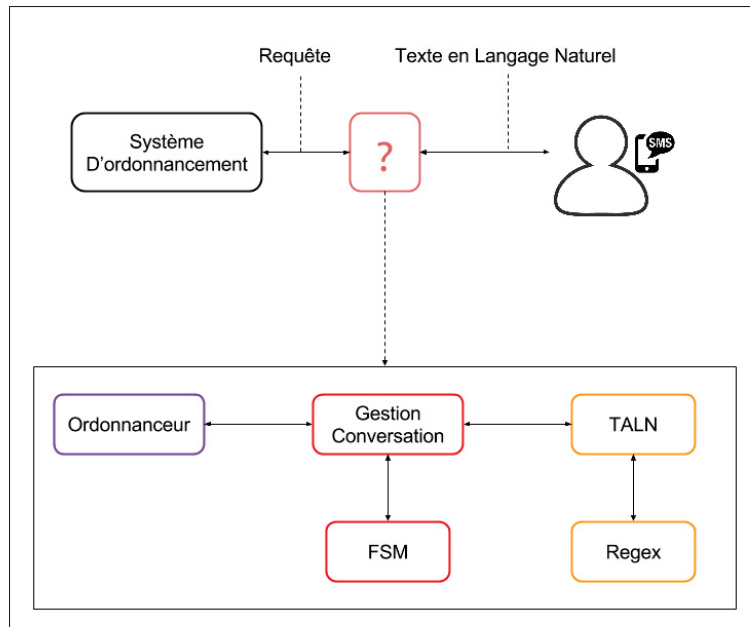


Figure 3.5 Décomposition du système en modules

3.4.1 Compréhension du langage naturel

Le module *TALN* regroupe l'ensemble des fonctions permettant l'extraction d'informations dans des textes en langage naturel et l'instanciation d'un objet de type *Message* (voir Section 3.3.2).

3.4.1.1 Extraction des services

L'extraction du service dans un message s'effectue à partir d'une liste de mots clés définissant un service dans le système. À partir de la liste des services pouvant être fournie par l'entreprise, nous avons déterminé un ensemble d'expressions en langage naturel pouvant les définir. Nous avons associé à chaque identifiant d'un service, dans le système d'ordonnancement, une ou plusieurs expressions en langage naturel. Ces expressions ont été déterminées à partir de la récolte de données (voir Chapitre 3). La liste étant restreinte, nous l'avons enrichie à partir de nos connaissances. De plus, étant donné que nous étudions les échanges par SMS, un ensemble de déclinaisons dans un langage SMS a dû être produit pour chaque expression.

Tableau 3.1 Description des modules

Module	Rôle
Gestion de la conversation	Réaliser les tâches de gestion : • interpréter un nouveau message dans le contexte d'une conversation ; • exécuter une tâche ; • déterminer la réponse à retourner. La fonction <i>GenerationReponse()</i> permet de générer une réponse à partir de l'état d'une conversation et de la réception d'un nouveau message, en utilisant les fonctions des modules <i>ordonnanceur</i>, <i>TALN</i> et <i>FSM</i>.
TALN	Effectuer le traitement d'un message en langage naturel : • extraire les expressions temporelles ; • extraire un service ; • extraire des mots clés.
Ordonnanceur	Interfacer avec l'API de prise de rendez-vous d'Agendize. Ce module regroupe un ensemble de fonctions utiles pour la gestion des rendez-vous, en particulier : • lister les services ; • vérifier une disponibilité ; • déterminer une liste de créneaux ; • enregistrer un rendez-vous ; • rechercher un client ; • enregistrer un client.
Machine à états (FSM)	Implémentation de la machine à états
Regex	Regroupe les expressions régulières représentant des configurations d'expressions temporelles dans les textes. Implémente un ensemble de règles pour le traitement des expressions régulières dans un texte et l'instanciation d'un objet de type Date (<i>voir</i> section 3.3.2).

3.4.1.2 Extraction des expressions temporelles

L'annotateur HeidelTime⁶ a été utilisé pour l'extraction des expressions temporelles dans les messages textes. Au cours de la phase préliminaire, l'utilisation de l'annotateur sur les messages récoltés avait permis d'identifier une liste d'erreurs et de limites (*voir* section 2.3.3.2).

Les améliorations à apporter concernait :

- le traitement des intervalles de temps ;
- la correction les dates avec des valeurs dans le passé ;
- l'association une heure à une date.

Le traitement des expressions temporelles se déroule en quatre étapes (*voir* Figure 3.6). D'abord le texte est annoté avec HeidelTime, produisant un fichier XML. Dans le texte d'origine, on remplace les expressions annotées d'une balise <TIMEX3> par une annotation (_DATE_ ou _TIME_) correspondant au type de l'expression. Pour finir, on utilise une liste d'expressions régulières pour reconnaître des expressions de temps particulières. Enfin, un objet de type *Date* est instancié à partir des expressions extraites, du type de l'expression régulière et d'une liste de règles d'association.

3.4.2 Gestion du dialogue

La gestion du dialogue est le module central du système. Il regroupe un ensemble de fonctions permettant la gestion de la conversation et la génération d'une réponse en langage naturel. La fonction principale *GenerationReponse()*, appelée par la vue (dans l'application), permet de générer une réponse à partir de l'état d'une conversation et de la réception d'un nouveau message. Nous présenterons d'abord les détails du processus de traitement d'une requête, suivi de l'implémentation de la machine à états et de la composition du message de réponse à partir de modèles.

6. Nous avons présenté cet outil dans le chapitre 2 (*voir* Section 2.3.3.1)

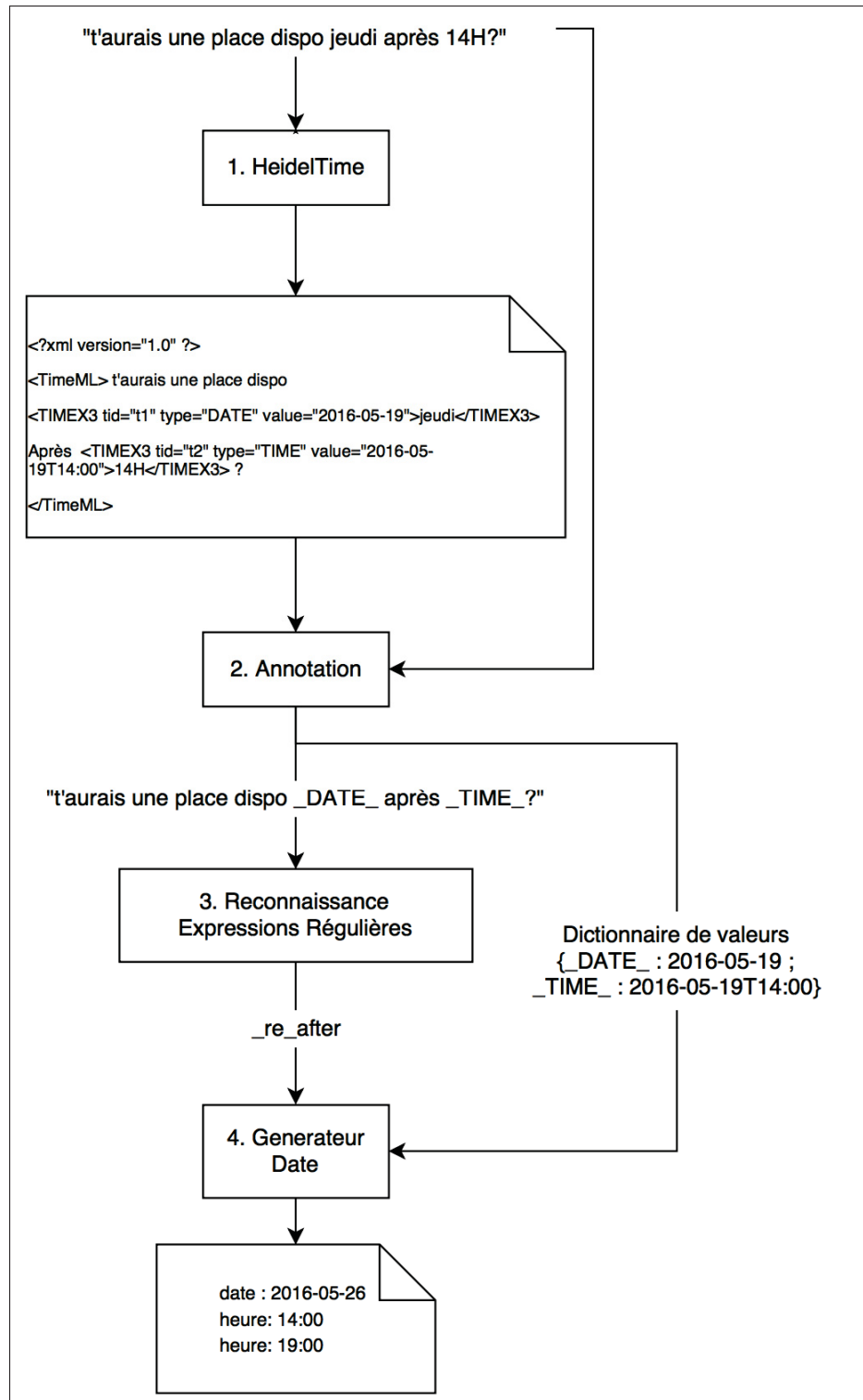


Figure 3.6 Extraction des expressions temporelles dans un texte

3.4.2.1 Traitement d'un message

Dans le diagramme d'activités (voir Figure 3.7) nous présentons le traitement effectué par la fonction *generationResponse()*. Lorsque la vue reçoit un message, les données de l'utilisateur, s'il existe, sont récupérées au format JSON. Si le client n'est pas connu dans le système, celui-ci est créé en BDD. Cette fonction est exécutée pour déterminer la réponse à retourner au client, en fonction de l'état de la conversation active et du message reçu. Lors du traitement, les informations sur le client sont modifiées et retournées, au format JSON, à la vue qui effectue une mise à jour des données en base de données.

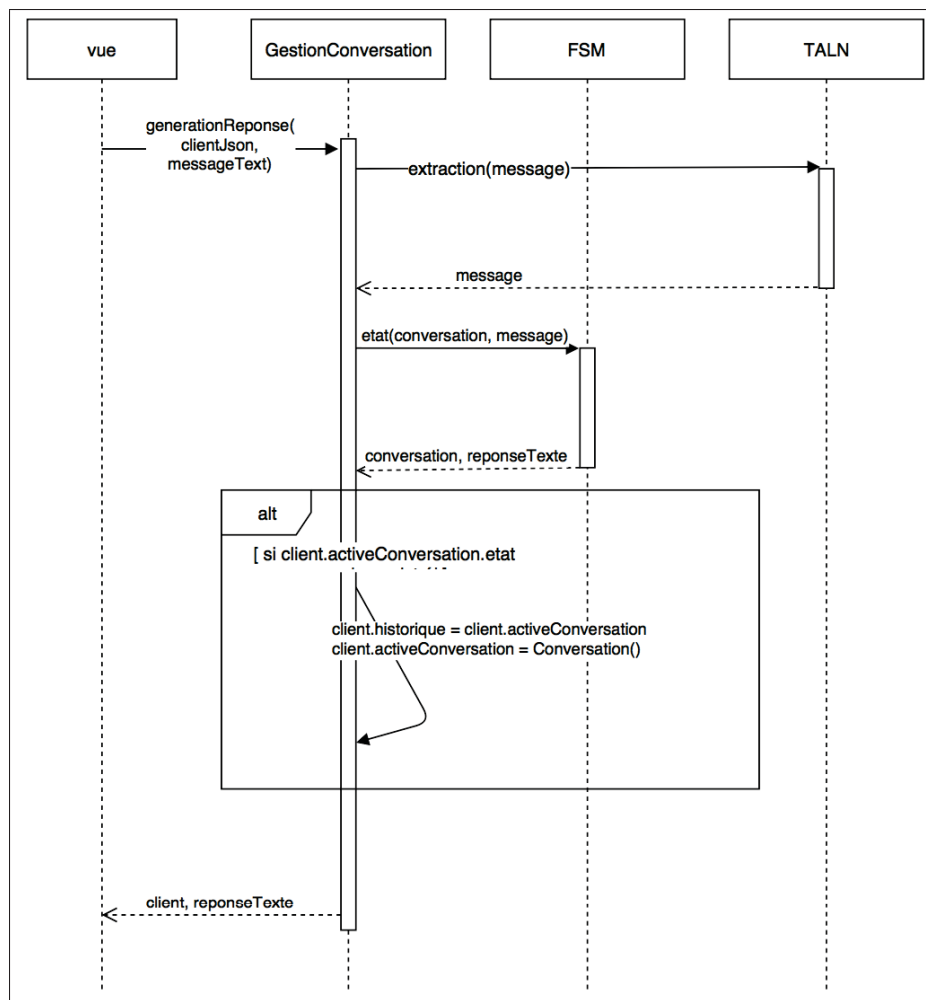


Figure 3.7 Génération de la réponse

D’abord, le module *TALN* permet d’extraire les informations du message et d’instancier un objet de la classe *Message*. Puis, la fonction correspondante à l’état de la conversation active est appelée, l’état de la conversation est mis à jour et la réponse est générée.

Si l’état *Enregistré* est atteint, le rendez-vous est enregistré et un identifiant unique est généré dans le système d’ordonnancement. De plus, la conversation est considérée comme terminée, elle est archivée dans l’historique et une nouvelle conversation est instanciée à l’état initial dans l’attente de la réception d’une nouvelle demande.

3.4.2.2 Développement de la machine à états

La machine à états est implémentée dans le module *FSM* dans lequel chaque état est représenté par une fonction prenant en paramètre une conversation et un message. Dans chaque état, des règles sont implémentées pour déterminer la réponse à retourner, l’action à effectuer et l’état suivant. Une fonction particulière peut être appliquée : *Traitement du rendez-vous*. Elle est appliquée si toutes les informations nécessaires sont recueillies et enregistrées dans la conversation. Elle permet d’effectuer une requête au système d’ordonnancement qui dépend du type de date traité (correspondant attribut *type*).

3.4.3 Génération de la réponse

Lors de la conception d’une interface conversationnelle la génération d’une réponse est une phase critique car ce sont ces réponses qui guident l’utilisateur. Les messages sont conçus manuellement à partir de modèles composés de variables. Nous avons identifié que la conception d’une réponse en langage naturel repose sur deux concepts : la précision et la compréhension. Pour créer ces modèles nous avons travaillé en collaboration avec l’*UX/UI designer*.

Nous avons déterminé qu’il était important de citer de manière explicite :

- les informations comprises par le système : « Vous avez demandé un brushing » ;
- la requête exécutée : « vous avez demandé un rendez-vous mardi entre 14h et 15h pour une coupe pour femme » ;

- le résultat de la requête (s'il y a lieu) : « le rendez-vous demandé est disponible » ou « j'ai plusieurs rendez-vous possibles » ;
- ce que le système attend en réponse : « J'ai besoin du service » ou « Répondez C pour confirmer ou M pour modifier ».

Finalement, nous avons déterminé quatre types de message de réponse :

1. La demande d'une information : « Tu souhaites prendre un rendez-vous le 20 mai à 15h pour une coupe. Je n'ai pas de disponibilité. Peux-tu me donner une nouvelle date ou un autre service ? » ;
2. La demande d'une confirmation : « Tu souhaites prendre un RV le 23 mai à 11h pour une coupe. Une place est disponible. Envoie C pour confirmer ou M pour modifier. Merci. » ;
3. La proposition d'un rendez-vous : « Tu n'as pas validé ton rendez-vous. Je peux te proposer le 25 mai à 10h. Envoie C pour confirmer ou M pour modifier. Ou bien, tu peux me donner une nouvelle date et/ou heure ou un autre service. Merci.» ;
4. La confirmation d'un rendez-vous : « Nous te confirmons ton RV pour le 25 mai à 14h Merci et Bonne journée. ».

3.4.4 Gestion de la prise de rendez-vous

Le module d'ordonnancement est l'interface de communication avec l'API du service d'ordonnancement. Il regroupe toutes les fonctions utiles pour le traitement des rendez-vous. Chaque fonction représente une requête HTTP au service d'ordonnancement. Le module est en charge de concevoir la requête, de l'exécuter et de traiter la réponse. Les fonctions principales sont :

- lister les créneaux disponibles : permet d'obtenir la liste des créneaux disponibles en fonction de critères donnés (une date de début, une date de fin et le service) ;
- vérifier la disponibilité d'un rendez-vous : pour une date, une heure et un service, la disponibilité du rendez-vous est vérifiée ;
- enregistrer un rendez-vous : enregistre le rendez-vous dans le système d'ordonnancement. La réponse du système est l'identifiant du rendez-vous.

3.5 Conclusion

3.5.1 Résultats

Dans cette section, nous présentons nos résultats sous la forme d'exemples d'interactions entretenues avec le prototype développé. Nous montrons en particulier les particularités identifiées concernant la prise de rendez-vous.

Le premier exemple (voir Figure 3.8) illustre le cas de base pour le traitement d'une requête. Quand toutes les informations sont obtenues, le système vérifie la disponibilité du rendez-vous demandé. S'il est disponible, une confirmation est demandée. S'il est accepté par le client, le rendez-vous est enregistré dans le système d'ordonnancement. La conversation est terminée et stockée dans l'historique des conversations. Cet exemple d'enchaînement de phases dans la conversation constitue le processus le plus simple à traitement. Cependant, la prise de rendez-vous n'est pas toujours aussi linéaire.

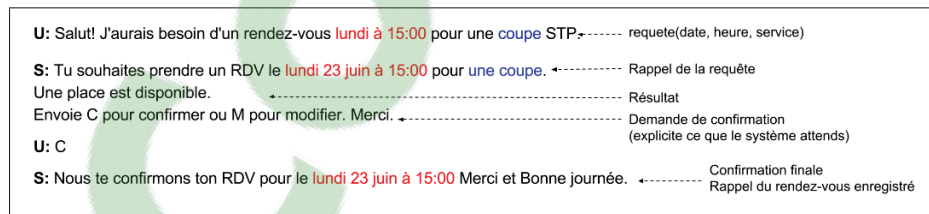


Figure 3.8 Exemple 1 : requête fermée

Un second exemple (voir Figure 3.9) présente le traitement d'une requête dite semi-ouverte⁷ où la notion de temps n'est pas précise (voir section 2.4.2). Plusieurs solutions sont envisageables, des précisions sont demandées ou bien un créneau est directement proposé au client. Dans le second exemple, un créneau était disponible dans un intervalle calculé à partir de l'expression temporelle extraite. Cependant, si ce n'est pas le cas, le système demande de nouvelles informations pour modifier la requête (voir Figure 3.10).

7. Nous avons donné une définition des différents types d'expressions temporelles dans le chapitre 3.

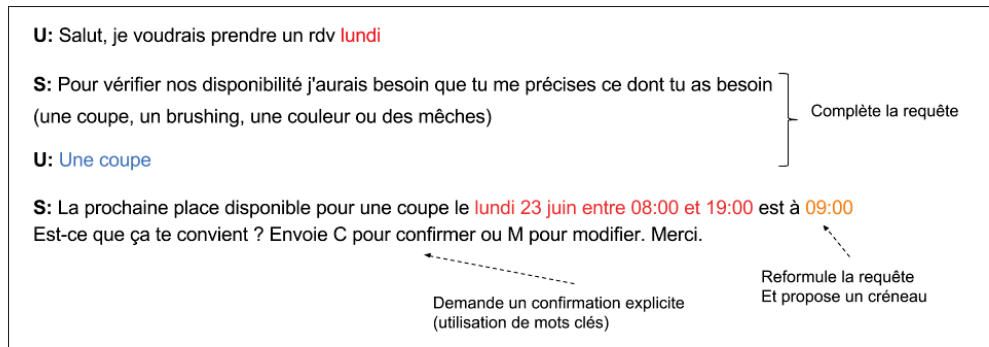


Figure 3.9 Exemple 2 : requête semi ouverte et proposition du système

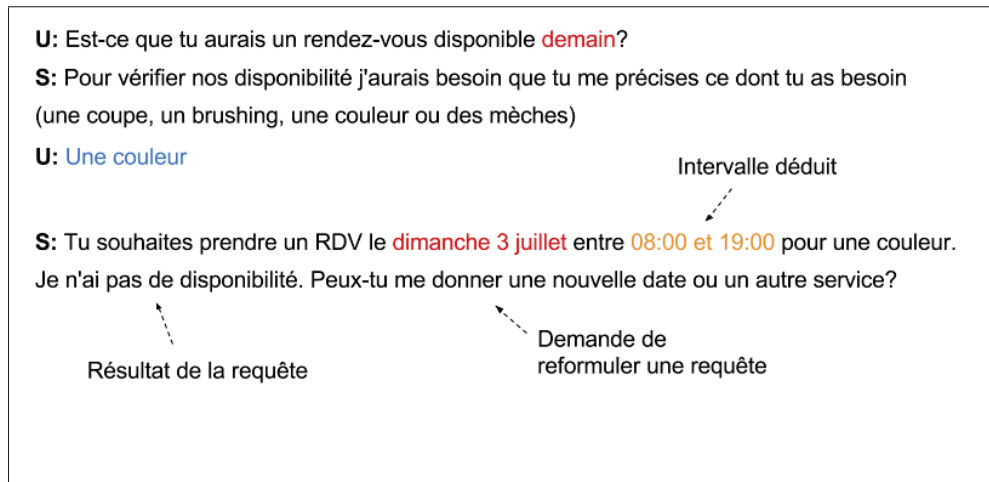


Figure 3.10 Exemple 3 : requête semi ouverte et modification des informations

Le client peut refuser une proposition en utilisant le mot clé précisé par le système. Alors, le système propose un nouveau créneau correspondant à la requête initiale et propose aussi à l'utilisateur de donner de nouvelles contraintes pour modifier sa requête. Si l'utilisateur donne une nouvelle information, la requête est modifiée et le résultat est de nouveau formulé (voir Figure 3.11).

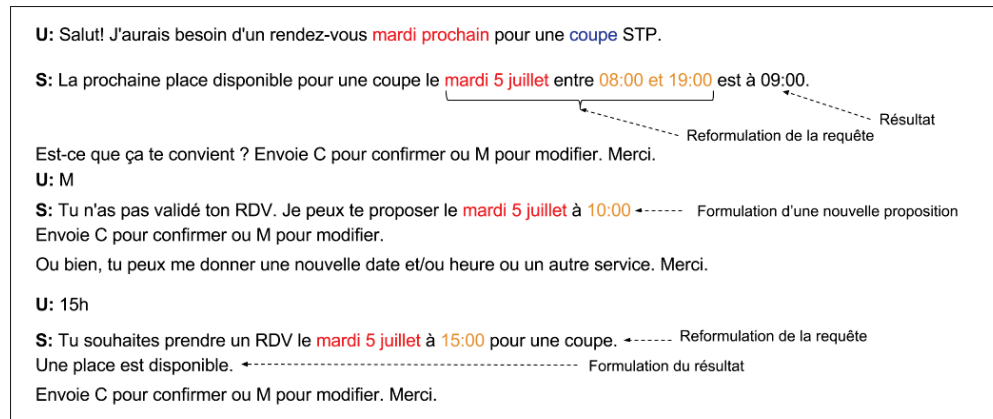


Figure 3.11 Exemple 4 : traitement du refus de l'utilisateur

3.5.2 Avantages et limites de la méthode

Initialement, nous avons fait l'hypothèse que pour être capable de réserver un rendez-vous, le système devait simplement récolter les informations (le service, la date et l'heure) requises, tel que réalisé dans le processus actuel (*voir* Figure 3.1). La machine à états permet de récolter ces informations, soit par extraction dans les messages du clients, soit par une demande explicite venant du système. La détermination d'une information supplémentaire nécessite l'ajout d'une étape dans l'extraction d'informations et d'un nouvel état dédié à cette dernière dans la machine à états (par exemple, nous pourrions demander le nom de l'employé).

3.5.2.1 La Gestion de la conversation

Contrairement à une approche basée sur une machine à états collectant les informations les unes à la suite des autres (présentée dans le chapitre 1), nous avons laissé plus de liberté à l'utilisateur. L'utilisateur initie la conversation en envoyant le premier message, il n'est donc pas contraint par les informations qu'il doit fournir. Mais la suite de la conversation est prise en charge par le système. L'utilisateur n'est pas en mesure de passer à l'étape suivante tant que l'information requise n'est pas fournie ou est erronée. L'approche considérée est suffisante tant que le nombre d'informations à récolter n'est pas trop important et que l'utilisateur effectue des requêtes précises. Comme nous l'avons présenté dans le cadre théorique la principale limite

d'une machine à états est l'augmentation du nombre d'états et de transition pour être capable de traiter des conversations plus complexe.

3.5.2.2 La compréhension de l'énoncé

Dans la phase préliminaire, nous avons proposé une catégorisation des différents messages : *demande* et *requête*. Dans ce prototype, nous nous sommes limités à la gestion d'une tâche précise, nous avons considéré que tous les messages étaient de même type : *requête(attribut,valeur)*.

La méthode de CLN utilisée semble pertinente pour reproduire le processus actuellement réalisé avec une interface graphique, mais insuffisante pour considérer les autres types de messages que nous avons identifiés dans la phase préliminaire (*voir* Section 2.4.2).

3.5.3 Proposition

La compréhension du langage naturel dans les textes pourrait être améliorée et enrichie, pour considérer des énoncés plus diversifiées, en étudiant la notion d'acte de dialogue (présentée dans le chapitre 1 : Cadre théorique).

Dans le chapitre suivant, nous présentons le deuxième prototype fondé sur l'identification d'actes de dialogue dans les messages. Pour cela, nous avons étudié les outils de conception d'interfaces conversationnelles proposés pour la conception d'Assistants Personnels Intelligents (IPA).

CHAPITRE 4

EXPLORATION D'UN OUTIL DE CONCEPTION D'INTERFACE CONVERSATIONNELLE

L'approche utilisée dans le premier prototype restreignait la tâche de CLN à l'extraction d'entités dans les textes. En utilisant une machine à états pour contrôler le dialogue nous avons conçu un système capable de gérer un processus de prise de rendez-vous. Cette approche, bien que pertinente pour la détermination d'une requête, ne permet pas de comprendre un ensemble d'énoncés plus variés (par exemple les demandes d'informations).

Les Assistants Personnel Intelligents (IPA) sont des systèmes de dialogue appliqués à l'exécution de tâches commandées par l'utilisateur au moyen d'interactions en langage naturel. Depuis le début des années 2000, ils accompagnent les utilisateurs dans leur quotidien grâce à leur intégration dans des appareils électroniques (téléphones mobiles, enceintes intelligentes, etc.) ou dans des outils de clavardage. Des plateformes de conception et de développement permettent aux développeurs de créer et de personnaliser leurs propres agents. La méthode repose sur la définition de compétences en associant des intentions (représentation de l'énoncé l'utilisateur) à une action exécutable par le système.

Le second prototype a utilisé l'un de ces outils pour appliquer les concepts associés aux IPA à notre contexte de prise de rendez-vous. **La section 1** présente la méthode et les outils utilisés. **Les sections 2 et 3** décrivent les phases de modélisation et d'implémentation. **La section 4** résume nos résultats, ainsi que les apports et limites de la méthode utilisée.

4.1 Méthodologie et matériel

4.1.1 Méthodologie

En considérant les limites du prototype précédent, nous nous sommes orientés vers l'utilisation d'un outil de conception d'interface conversationnelle¹. Par conséquent, les méthodes utilisées pour chacune des tâches du système de dialogue sont les suivantes :

- **Contrôle du dialogue :**

Le contrôle du dialogue repose sur une machine à états et l'application de la notion de formulaire². En effet, une intention est décrite comme un formulaire avec un ensemble d'entités obligatoires ou non. L'action à exécuter est déterminée à partir de l'intention (reconnue dans l'énoncé) et de l'état actuel de la conversation. Une machine à état est aussi utilisée pour compléter les informations manquantes avant d'exécuter une action.

- **Compréhension du langage naturel :**

La représentation sémantique du message est de la forme *intention(attribut,valeur)*. L'intention est déterminée à l'aide d'un algorithme d'apprentissage automatique intégré dans l'outil que nous utilisons. L'extraction des attributs se concentre sur les expressions temporelles et les services.

- **Génération de la réponse :**

Les réponses sont générées à partir de modèles de messages composés de variables pour lesquelles une valeur est déterminée en fonction du contexte.

La phase de modélisation s'est concentrée sur l'identification des intentions, des actions et des états, à partir des observations antérieures. De plus, nous avons revu la modélisation des données du système. Tout en conservant une architecture identique au premier prototype, nous avons adapté et amélioré les fonctions de chacun des modules.

1. L'entreprise partenaire souhaitait étudier la pertinence de cet outil aux besoins de la tâches.

2. Ces méthodes sont définies dans la section 1.2.3.

4.1.2 Choix technologiques

A partir des applications Web présentées dans les chapitres précédents (*voir* chapitre 2 et Chapitre 3) nous avons conçu une application recevant des SMS du service Twilio, les traitants et retournant une réponse à l'expéditeur par l'intermédiaire de ce même service. Le flux des échanges entre l'application et les services utilisés est illustré dans la Figure 4.1.

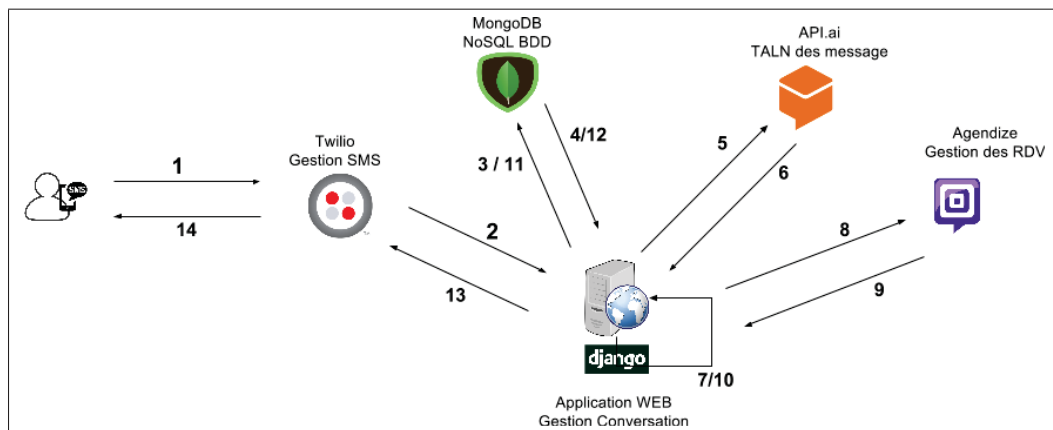


Figure 4.1 Architecture de l'application

Nous avons fait le choix d'utiliser la plateforme *api.ai*³ pour le développement de notre nouveau prototype. Cette plateforme à l'avantage d'offrir une librairie python s'intégrant parfaitement à notre développement actuel.

4.2 Modélisation

Dans cette section, nous décrivons la phase de modélisation de la conversation et des données utilisées dans l'application. La modélisation de la conversation consiste en la description des intentions reconnaissables et des actions exécutables par le système ainsi que des transitions entre les différents états.

3. <http://api.ai>

4.2.1 Modélisation de la conversation

4.2.1.1 Définition des intentions

À partir de nos connaissances de la conversation que nous souhaitons modéliser et des conclusions effectuées à la suite des deux étapes précédentes (*voir* chapitres 2 et 3), nous avons déterminé un ensemble de paires intention-action. Le tableau 4.1 décrit les intentions considérées.

Tableau 4.1 Définition des intentions

Intention	Définition
Contact	L'utilisateur est à l'initiative de la conversation, il envoie le premier message qui amorce la conversation. Cette intention représente les messages qui ne sont ni des <i>requêtes</i> , ni des <i>demandes</i> .
Réserve	Demande d'un rendez-vous (corresponds la catégorie <i>requête</i>)
Demande disponibilités	Demande les disponibilités de l'entreprise (corresponds à un cas particulier de la catégorie <i>demande</i>)
Confirme	Une affirmation positive
Refuse	Une affirmation négative
Autre	Des messages hors contexte pour le système

4.2.1.2 Définition des actions

Les actions représentent les compétences de l'agent conversationnel. Une action peut avoir un effet sur la conversation (produire une réponse) et (ou) sur l'état interne du système. Nous présentons dans le tableau (*voir* Tableau 4.2) les actions possibles et leurs définitions. Pour chaque action, un exemple d'énoncé de réponse est fourni.

L'action *Accueillir* est exécutée lors de la réception du premier message s'il ne représente ni une requête, ni une demande d'information prise en charge par le système. En effet, l'utilisateur est à l'initiative de la conversation et envoie le premier message. Dans le message d'accueil, le système doit se présenter et demander à l'utilisateur les informations dont il a besoin. Les

Tableau 4.2 Définition des actions.

Action	Définition	Énonce
Accueillir	Présentation des fonctionnalités et des informations attendues.	Bonjour, je suis le système de prise de rendez-vous. Peux-tu me donner une date et un service (liste des services) pour que je puisse vérifier mes disponibilités ?
Vérifier une disponibilité	Vérification des disponibilités d'un rendez-vous.	Le rendez-vous que vous avez demandé pour <i>DATE</i> à <i>HEURE</i> pour <i>SERVICE</i> est bien disponible.
Réserver	Enregistrement du rendez-vous dans le système d'ordonnancement. Mise à jour de la conversation avec l'identifiant du rendez-vous. Confirmation de l'enregistrement du rendez-vous à l'utilisateur.	Votre rendez-vous pour le <i>DATE</i> à <i>HEURE</i> pour <i>SERVICE</i> a bien été confirmé par notre entreprise. Au plaisir de vous voir. Bonne journée.
Faire une proposition	Proposition d'un rendez-vous et demande de confirmation.	Je peux vous proposer un rendez-vous pour le <i>DATE</i> à <i>HEURE</i> pour <i>SERVICE</i> . Est-ce que cela vous conviendrait ?
Répéter	Traitement des messages qui ne sont pas compris par le système. Répétition de l'énoncé précédent.	Je n'ai pas compris votre dernier message. J'attendais une confirmation de votre part pour le <i>DATE</i> à <i>HEURE</i> pour <i>SERVICE</i> .

échanges par messages courts restreignent l'explication donnée, le texte doit être clair et concis pour guider l'utilisateur.

L'action *Vérifier une disponibilité* interroge le système d'ordonnancement pour vérifier les disponibilités d'un rendez-vous demandé. Le résultat de la requête est retourné en langage naturel.

L'action *Réserver* enregistre le rendez-vous final, confirmé par le client, dans le système d'ordonnancement (correspond au traitement exécuté dans l'état Enregistré du premier prototype).

L'action *Faire une proposition* détermine, à partir d'une liste de disponibilités de l'entreprise, une proposition à faire au client. Pendant la phase préliminaire et à la suite du développement du premier prototype, nous avons montré que le système était un acteur de la conversation et qu'il représentait l'entreprise impliquée dans la prise décision (pour le choix d'une date et d'une heure). Le client effectue une demande, mais si celle-ci est refusée deux stratégies sont envisageables. La première est de demander au client de reformuler un autre choix mais ceci peut déboucher à de longs échanges. La seconde stratégie suppose de faire faire une proposition pour tenter d'aboutir plus rapidement à une date et à une heure satisfaisantes pour les deux parties.

L'action *Répéter* est une action indispensable dans le cas de la gestion d'une conversation. En effet, le système est conçu pour comprendre un ensemble d'énoncés prédéfinis lui permettant d'atteindre un but. Cependant, les énoncés qui ne sont pas compris doivent quand même être considérés par le système. L'utilisateur est orienté par des questions précises et explicites sur les informations nécessaires pour atteindre le but. Nous avons convenu que lorsqu'un énoncé n'était pas compris, la dernière réponse envoyée par le système serait répétée.

4.2.1.3 Modélisation de la machine à états

Le dialogue est géré par une machine à états (*voir* Section 1.2.3) défini par le quintuple $(\Sigma, S, s_0, \delta, F)$. Les transitions dépendent de l'état courant et de l'intention reconnue (*voir* Tableau 4.3).

L'ensemble des états S est :

- *aucun* est l'état initial s_0 , où aucun échange n'a encore été effectué ;
- *accueilli* représente l'état où un échange a déjà été effectué, c'est à dire que le système a déjà eu l'occasion de se présenter. Cet état nous permet de contextualiser le message retourné au client. Par exemple, pour l'action *vérifier disponibilité* dans le message de réponse on pourrait ajouter une phrase introductive (« bonjour ... »), si le client n'a pas encore été accueilli ;

- *a confirmer* correspond à l'état où une proposition a été faite à l'utilisateur et où le système attend une confirmation de sa part ;
- *enregistre* est l'unique état final de F . Il est atteint lorsque qu'une confirmation est reçue.

Le tableau 4.3 présente les transitions entre les états et associe une intention (ligne) à une action (rouge) en fonction de l'état courant (colonnes).

Tableau 4.3 Paires intention-action et transitions entre les états

	AUCUN	ACCUEILLI	A CONFIRMER
CONTACT	accueillir ACCUEILLIR	répéter A CONFIRMER	répéter A CONFIRMER
RESERVE	Vérifier une disponibilité A CONFIRMER	Vérifier une disponibilité A CONFIRMER	Vérifier une disponibilité A CONFIRMER
DEMANDE	Faire une proposition A CONFIRMER	Faire une proposition A CONFIRMER	Faire une proposition A CONFIRMER
CONFIRME	accueillir ACCUEILLI	répéter ACCUEILLI	réserve ENREGISTRE
REFUSE	accueillir ACCUEILLI	répéter ACCUEILLI	Faire une proposition A CONFIRMER
AUTRE	accueillir ACCUEILLI	répéter ACCUEILLI	répéter A CONFIRMER

Les paires intention-action sont représentées par des formulaires pour lesquels des informations peuvent être requises. Afin d'être en mesure d'exécuter l'action, le système fait une demande explicite des informations requises. Le processus est modélisé par une machine à états secondaire, où les états représentent les informations à fournir et les transitions représentent les réponses de l'utilisateur. Dans chaque état, la fonction de transition est une question du système sur l'information manquante.

4.2.2 Modélisation des données

Comme pour le prototype précédent, un client a un identifiant unique déterminé dans le système d'ordonnancement, au moins une conversation active et peut avoir un historique composé de l'ensemble des conversations antérieures (voir figure 4.2).

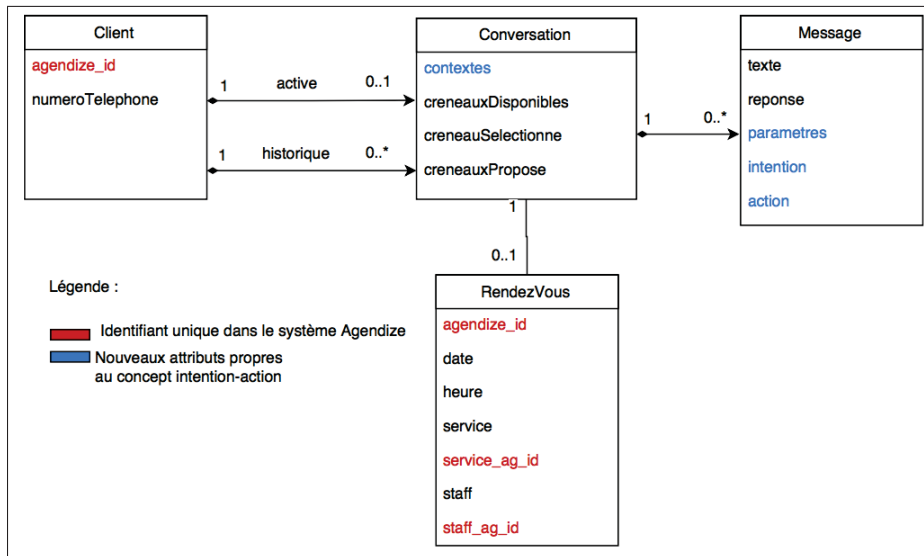


Figure 4.2 Modèle des données

Une conversation est composée de plusieurs messages. Dans un message, l'attribut *parametres* représente les entités extraites du message. L'attribut *action* représente l'action à exécuter par le système en fonction de l'état (*contextes*) de la conversation et de l'intention identifiée. Enfin, l'attribut *réponse* est l'énoncé en langage naturel retourné par le système pour ce message.

Un rendez-vous est un élément unique de la conversation. Les attributs *date*, *heure* et *service* sont extraits des messages alors que *service_ag_id* et *staff_ag_id* sont déterminés par le système d'ordonnancement. Un rendez-vous est lié à une conversation et est identifié une fois confirmé et validé par son identifiant dans le système d'ordonnancement *agendize_id*.

L'objectif de la conversation est de déterminer un rendez-vous entre le client et l'entreprise. Au cours de la conversation, le système est amené à faire des propositions de rendez-vous.

L'attribut *creneauxDisponibles* est une liste de disponibilités (une date et une heure) obtenue en effectuant une requête auprès du système d'ordonnancement. Le *creneauSelectionne* est l'index dans la liste des disponibilités de la proposition effectuée (s'il y a eu une proposition). L'attribut *creneauxPropose* est une liste de créneaux déjà proposés au cours de la conversation. L'attribut *contextes* est une liste de contextes représentant l'état de la conversation pour exécuter des requêtes à l'outil de CLN.

4.2.3 Décomposition en modules

Nous avons divisé les fonctions de l'application en trois modules principaux (*voir* Figure 4.3) ayant chacun un rôle dans le traitement de la conversation (*voir* Tableau 4.4).



Figure 4.3 Décomposition en modules

4.3 Implémentation

Dans cette section nous détaillons l'implémentation des modules introduits dans la section 4.2.3. Nous présentons les moyens employés pour la compréhension du langage naturel, par la reconnaissance d'intentions dans les textes, ainsi que le traitement effectué par le module de gestion de la conversation. Le module d'ordonnancement n'est pas mentionné, car aucune modification n'a été apportée.

Tableau 4.4 Description des modules

Module	Rôle
TALN	• traitement du langage naturel dans les textes ; • reconnaissance d'une intention dans un texte ; • extraction de la date, l'heure et le service.
Gestion de la conversation	• exécuter une action ; • mettre à jour l'état de la conversation (le contexte et les informations) ; • générer des réponses en langage naturel.
Ordonnanceur	Interfacer avec l'API de prise de rendez-vous : • chercher une liste de disponibilité ; • vérifier une disponibilité ; • enregistrer un rendez-vous ;

4.3.1 Compréhension du langage naturel

4.3.1.1 Création de l'ensemble d'entraînement

La reconnaissance d'une intention est une tâche de classification. La classification est une méthode d'apprentissage supervisé, qui utilise un ensemble de données, nommé d'entraînement, pour déterminer un modèle. Pour construire notre ensemble de données d'entraînement nous avons utilisé des phrases extraites de la phase préliminaire (*voir* Chapitre 3).

La figure 4.4 présente un exemple de phrases, en langage naturel, annotées avec l'intention *Réserver*. Afin d'enrichir l'ensemble de données, nous avons aussi utilisé des modèles de phrases. Dans chaque énoncé, des entités sont extraites (*DATE*, *HEURE*, *SERVICE*), à partir de celles-ci nous avons pu formuler des modèles d'expressions en langage naturel.

Textes en langage naturel extrait de la récolte :

« Hello. Une coupe ce soir 18h »

« J'aimerais me faire couper les cheveux le 24 août prochain à 13h30 »

« Bonjour, j'aimerais prendre un rendez-vous pour me faire couper les cheveux. êtes-vous disponible ? »

Modèles d'énoncés :

« DATE à HEURE pour SERVICE »

« HEURE le DATE »

« SERVICE le DATE »

Figure 4.4 Ensemble d'exemples de l'intention *réserver*

4.3.1.2 Reconnaissance d'intention dans les messages

La plateforme utilisée pour la conception de l'agent et la définition des intentions met à disposition une API *REST* avec laquelle l'application communique par l'intermédiaire de requêtes *HTTP*. Le module TALN regroupe un ensemble de fonctions pour l'exécution de requête *HTTP*. Lorsque l'application reçoit un message, elle effectue une requête *POST* à l'API. Les paramètres de la requête sont le message texte et l'état courant de la conversation. La réponse au format JSON contient, entre autres, l'intention et les entités extraites dans le texte et l'action à exécuter.

4.3.2 Gestion de la conversation

Le traitement d'un nouveau message entrant s'effectue selon les étapes présentées dans le diagramme 4.5. D'abord, l'action à exécuter est déterminée à partir de l'intention reconnue et de l'état actuel de la conversation. Pour certaines actions des paramètres sont nécessaires. Par exemple, dans le cas de la vérification d'une disponibilité, une date, une heure et un service sont requis pour exécuter l'action. Si le système ne détient pas déjà toutes les informations, alors il demande explicitement chacune des informations à l'utilisateur dans un ordre prédéterminé. Dans le cas contraire, les données de la conversation sont mises à jour et l'action est exécutée. L'ensemble des actions est présenté dans le tableau 4.2. Dans certains cas, une requête est exécutée auprès de l'ordonnanceur. Enfin, en fonction du résultat de l'action et de l'état de la

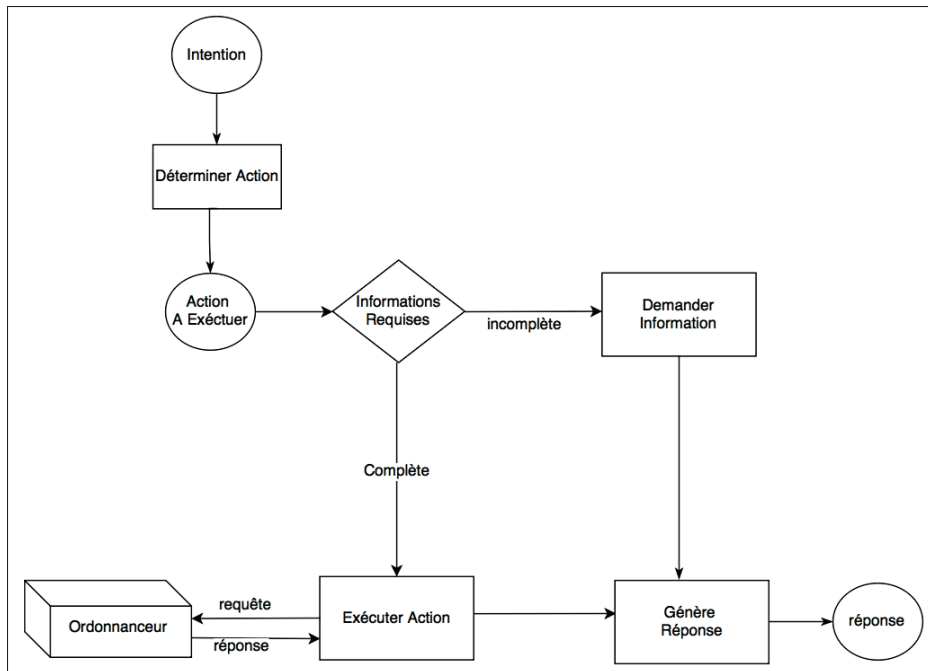


Figure 4.5 Catégorisation des messages

conversation une réponse en langage naturel est construite à partir de modèles, enregistrée et retournée à l'utilisateur.

4.4 Conclusion

4.4.1 Résultats

4.4.1.1 Exemples d'interactions

Dans cette section, des exemples de conversations élaborées sur le modèle et le développement présentés précédemment sont décrites. Ces conversations mettent en évidence la notion d'intention et l'exécution d'actions qui en découle.

Le premier exemple (*voir* Figure 4.6) illustre un échange similaire au premier prototype. La prise de rendez-vous est réalisée en trois phases : le système complète les informations dont il

a besoin, effectue une requête auprès de l'ordonnanceur et demande une confirmation auprès du client.

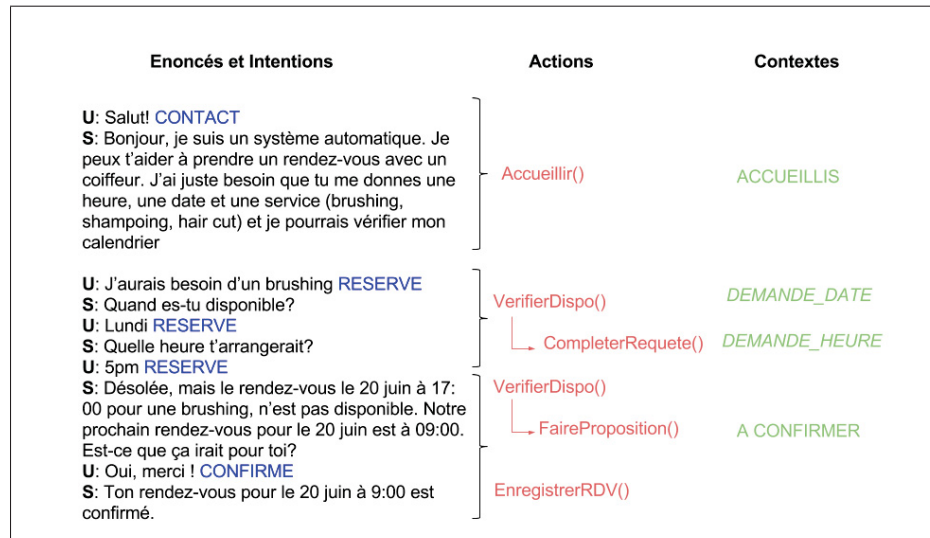


Figure 4.6 Exemple 1 : gestion d'une requête simple

Dans le second exemple (voir Figure 4.7), la phrase d'introduction est identifiée comme l'intention *RESERVE*. Le système n'utilise pas de phrase d'introduction et demande directement une date et une heure pour le service demandé. Dans cet exemple, le refus du client est illustré. La stratégie utilisée, qui consiste à proposer le prochain rendez-vous, conduit à effectuer plusieurs échanges pour satisfaire la requête du client.

Le dernier exemple (voir Figure 4.8) illustre la demande de disponibilité dépendante du service demandé et pour laquelle le système fait une proposition.

4.4.1.2 Stratégie d'entreprise

La gestion du dialogue est une tâche complexe dans les SD. Nous avons observé que dans un contexte de prise de rendez-vous la difficulté de conception reposait sur le nombre d'échanges nécessaires pour converger vers un rendez-vous final. La requête d'un utilisateur n'est pas toujours complète et le créneau demandé n'est pas nécessairement disponible. Les échanges

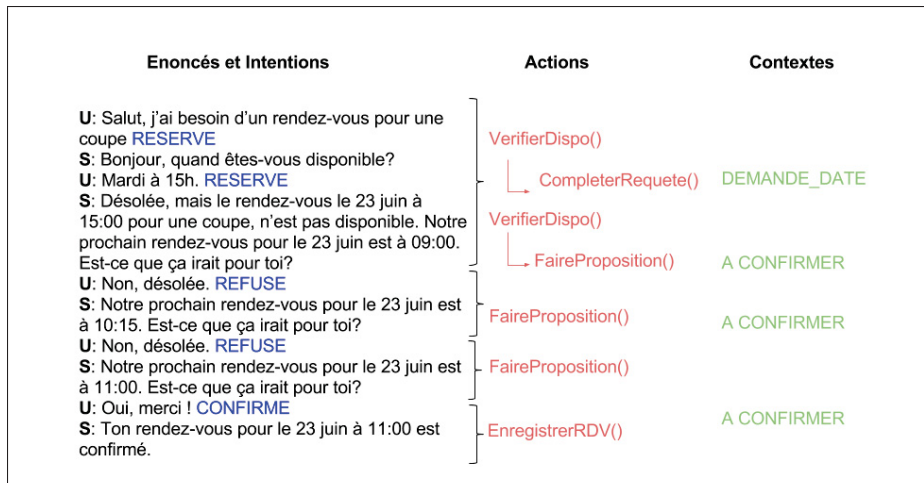


Figure 4.7 Exemple 2 : gestion d'un refus

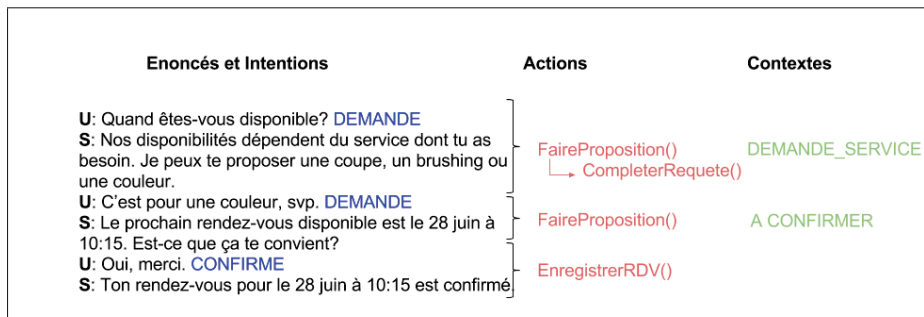


Figure 4.8 Exemple 3 : demande de disponibilités

entretenus à l'aide du prototype ont souligné l'importance de la formulation d'une proposition pertinente sur la gestion du dialogue (illustrée dans l'exemple 4.7).

Dans la littérature, nous avons étudié la gestion du dialogue dans la prise de rendez-vous en termes d'acte de négociation et de collaboration pour la détermination d'une date et d'une heure. Pendant les échanges effectués entre un client et une entreprise, le choix d'une proposition pertinente ne repose pas seulement sur la sélection d'un créneau dans une liste de disponibilités, mais sur la stratégie employée par l'entreprise.

Cette stratégie est spécifique à chaque type d'entreprise et est très variable. Elle peut dépendre de contraintes temporelles. Certaines entreprises vont préférer concentrer leurs rendez-vous

autour d'une certaine heure de la journée ou d'une journée particulière. Mais elle dépend aussi du degré d'importance accordé aux contraintes du client et de l'entreprise. Certaines entreprises vont avoir un vaste choix de créneaux possibles (par exemple les coiffeurs) ce qui permet de valoriser les contraintes des clients. Le profil du client peut aussi influencer sur ce choix. Alors qu'au contraire, d'autres entreprises ont des disponibilités plus restreintes (par exemple les services médicaux), ce sont alors les contraintes de l'entreprise qui comptent.

4.4.2 Apports et limites

L'objectif de ce prototype était de rendre le système plus flexible en permettant de comprendre des énoncés plus diversifiés. L'exploration d'une plateforme de développement d'agents personnels intelligents nous a permis de revoir la conception de notre système de dialogue. D'une part, nous avons amélioré la phase de compréhension du langage naturel en classifiant les énoncés pour déterminer des intentions. D'autre part, nous avons utilisé une méthode fondée sur la notion théorique de formulaires et une machine à état pour gérer le flux du dialogue.

Nous avons modélisé la conversation et les données à partir de nos connaissances et nos objectifs. Le prototype actuel a donc été construit sur une idée précise de la conversation que nous souhaitons entretenir. Bien que nous ayons utilisé des modèles de phrases, le manque de données a restreint les énoncés possibles.

Bien que nous nous soyons orientés vers un système de dialogue dirigé vers une tâche précise, nous avons pu observer les limites des approches basées sur des machines à états et la modélisation par formulaires. En effet, les Assistants Personnels Intelligents sont conçus pour exécuter des tâches très précises. Une fois la tâche réalisée, la conversation est terminée. Or, dans le cas de la prise de rendez-vous, l'exécution d'une action n'amène pas directement à la solution et à la conclusion de la conversation.

CONCLUSION

Nous présentons dans cette section, nos conclusions sur les travaux réalisés selon deux axes :

- un retour sur les questions à l'origine du projet et les objectifs que nous nous étions fixés ;
- une présentation des directions futures envisagées par l'entreprise pour l'amélioration de leurs services.

Retour sur les questions préliminaires et les objectifs :

L'étude exploratoire menée sur l'intégration d'une interface conversationnelle dans un système de prise de rendez-vous visait à identifier :

- l'effet du changement du moyen de communication entre un utilisateur et un système sur la gestion de la prise de rendez-vous ;
- les caractéristiques particulières d'une conversation pour la prise de rendez-vous.

L'étude s'est reposée sur le développement d'un prototype d'interface conversationnelle intégrée au service de prise de rendez-vous actuellement développé par l'entreprise partenaire.

La phase préliminaire de l'étude visait à récolter des messages auprès d'utilisateurs. Trois tâches ont été réalisées sur les données récoltées : l'identification de concepts, l'analyse des expressions temporelles et la catégorisation de l'ensemble de messages. Au contact des utilisateurs, nous avons fait les constats suivants. Alors que l'objectif est de rendre l'interaction plus naturelle l'utilisateur est en demande d'instructions venant du système. Bien que les interfaces conversationnelles aient l'ambition de faciliter l'interaction avec un système, les utilisateurs peuvent rapidement être découragés par des systèmes encore trop peu performants. Ils préfèrent toujours utiliser des interfaces graphiques traditionnelles ou chercher à contacter un humain s'ils ne sont pas satisfaits par l'échange réalisé avec le système conversationnel.

Le premier prototype conçu à partir d'une machine à états, gère des interactions simples et pré-définies. Il prend en charge des requêtes et peut, dans une certaine mesure, conclure sur un rendez-vous. Le second prototype s'est reposé sur l'utilisation d'un outil de conception d'interfaces conversationnelles pour les IPA.

L'application de ces outils à notre étude du processus de prise de rendez-vous a montré certaines limites. En effet, si les informations sont complètes et structurées le système est capable d'atteindre son but. Néanmoins, la prise de rendez-vous est un processus qui peut s'étaler dans le temps, être incomplet et imprécis même si finalement on vise la détermination d'une date et d'une heure précises.

Ces expériences ont mis en avant l'effet de l'utilisation d'échanges en langage naturel et asynchrone (impliquée par l'utilisation de SMS) sur la gestion de la prise de rendez-vous. D'une part, un rendez-vous peut être incomplet et se construit pendant la conversation. Le rendez-vous passe par différents états à partir de l'initialisation de la conversation jusqu'à la confirmation d'un rendez-vous final et complet. D'autre part, contrairement à une interface graphique, l'utilisation d'une interface conversationnelle limite la quantité d'informations que l'on peut communiquer à l'utilisateur. Par exemple, avec une interface graphique, les disponibilités sont affichées dans un calendrier dans lequel l'utilisateur sélectionne le créneau qui lui convient. La proposition d'un rendez-vous se restreint à retourner une liste de disponibilités selon les critères sélectionnés par l'utilisateur. Cependant dans notre contexte, lister toutes les disponibilités n'est pas envisageable, un sous ensemble de créneaux doit être sélectionné parmi cette liste. La proposition d'un rendez-vous devient une tâche critique et centrale dans la gestion de la prise de rendez-vous.

Un système est dit utilisable lorsque l'utilisateur est capable d'atteindre son but de manière efficace et satisfaisante. En utilisant des interfaces conversationnelles, la fonctionnalité du système repose sur sa capacité à atteindre son but dans un nombre d'échanges acceptables par l'utilisateur. La convergence rapide permet de garder le contact avec l'utilisateur, de maintenir la conversation et de maximiser sa satisfaction. Nous avons identifié que la convergence vers un rendez-vous final satisfaisant dépendait de la pertinence de la proposition effectuée. Néanmoins, ceci dépend des contraintes du client mais surtout de la capacité du système à reproduire le comportement d'une entreprise pour attribuer un rendez-vous.

Notre réflexion s'est portée sur la notion de stratégie d'entreprise pour l'attribution d'un rendez-vous. Celle-ci représente les règles, plus ou moins formelles, appliquées par une entreprise pour sélectionner un rendez-vous qui puisse, à la fois satisfaire les contraintes du client et de l'organisation. Cependant, la conception d'un système intelligent reproduisant ce comportement implique de pouvoir énumérer ou déduire ces règles qui sont propres à chaque organisation. Dans certaines situations, les contraintes de l'entreprise seront plus importantes que celles du client, en particulier lorsque la quantité de créneaux est faible. Par exemple, pour les services médicaux, c'est souvent le client qui se plie aux contraintes imposées par l'entreprise. En revanche, pour des secteurs comme les coiffeurs, l'ensemble des disponibilités étant plus large la proposition est plus ouverte aux contraintes du client.

L'étude des données et les deux phases de développement du prototype ont permis de mettre en avant l'étendue des domaines et des travaux à réaliser pour construire un système complet. La gestion de la prise de rendez-vous va au delà de la tâche d'ordonnancement de calendriers actuellement réalisée. Dans la section suivante, nous présentons une proposition pour les futurs travaux de l'entreprise.

Directions futures pour l'entreprise partenaire :

Une intégration future des conclusions précédentes nécessiterait des transformations dans le système d'information actuel pour intégrer la modélisation :

- d'un rendez-vous incomplet ;
- des préférences d'un client ;
- de la stratégie d'une entreprise de service pour l'attribution de rendez-vous.

La prise en charge de différents états pour un rendez-vous n'est pas considérée dans le système actuel. En plus d'être indispensable pour le développement d'un outil conversationnel, l'état d'un rendez-vous est une information qui semble pertinente pour la détermination d'une proposition. Cette dernière pourrait considérer les préférences du client déduites à partir des différents types de rendez-vous antérieurs (en cours de construction, proposé, demandé, enregistré ou annulé).

D'autre part, la proposition d'un rendez-vous repose sur une stratégie d'entreprise. Chaque entreprise a des préférences et applique un ensemble de règles pour formuler une proposition. Un autre axe de recherche serait de reproduire cette stratégie pour l'appliquer dans la prise de décision. La pertinence et l'efficacité d'une fonctionnalité pour effectuer une proposition pourraient être évaluées auprès des utilisateurs en l'ajoutant à l'outil de prise de rendez-vous en ligne existant.

Concernant le développement d'un outil conversationnel, l'étude menée a montré la complexité de la conception d'un outil automatisé gérant des interactions satisfaisantes pour l'utilisateur. Actuellement, les recherches évoluent mais le passage de la recherche à des outils commerciaux est encore limité et demande une mise à disposition de ressources (humaines, données et monétaires) conséquentes. Une première étape pourrait consister à aider les entreprises dans leur processus prise de rendez-vous en prenant en charge une partie des requêtes des clients et en les redirigeant par la suite vers un humain.

L'un des manques que nous avons souligné concerne les données permettant d'utiliser des méthodes plus sophistiquées. L'entreprise partenaire propose déjà un outil de clavardage intégrable à des sites internet pour permettre à un utilisateur de converser en temps réel et par messages instantanés avec un employé (à des fins de support client). Afin de récolter des dialogues entretenus dans le domaine étudié, l'outil pourrait être utilisé. Des entreprises pourraient proposer à leurs clients de prendre des rendez-vous par clavardage, en mettant à disposition les ressources humaines nécessaires. Par la suite, un outil semi automatisé pourrait y être intégré et la fonctionnalité de proposition automatique pourrait être évaluée. Le système de dialogue serait capable de prendre en compte un ensemble de requêtes. Si les demandes des utilisateurs sont incompréhensibles ou si la conversation ne converge pas dans une limite d'échanges donnée alors l'utilisateur pourrait être redirigé vers un opérateur humain. L'ensemble des échanges permettrait de construire un ensemble de données plus conséquent et robuste.

BIBLIOGRAPHIE

- Allen, James, Mehdi Manshadi, Myroslava Dzikovska, et Mary Swift. 2007. « Deep linguistic processing for spoken dialogue systems ». In *Proceedings of the Workshop on Deep Linguistic Processing*. p. 49–56. Association for Computational Linguistics.
- Allen, James F et C Raymond Perrault. 1980. « Analyzing intention in utterances ». *Artificial intelligence*, vol. 15, n° 3, p. 143–178.
- Aust, Harald, Martin Oerder, Frank Seide, et Volker Steinbiss. 1995. « The Philips automatic train timetable information system ». *Speech Communication*, vol. 17, n° 3, p. 249–262.
- Austin, John L., 1962. *How to do things with words*. Cambridge : Harvard University Press.
- Austin, John L., 1975. *How to do things with words*, volume 120. Cambridge : Harvard University Press.
- Banchs, Rafael E. et Haizhou Li. 2012. « IRIS : a chat-oriented dialogue system based on the vector space model ». In *Proceedings of the ACL 2012 System Demonstrations*. p. 37–42. Association for Computational Linguistics.
- Bittar, André, Pascal Amsili, et Pascal Denis. 2011. « French TimeBank : un corpus de référence sur la temporalité en français ». In *TALN 2011-Traitement Automatique des Langues Naturelles*. p. 259–270. Laboratoire d'Informatique de Robotique et de Micro-électronique.
- Bonneau-Maynard, Hélène, Matthieu Quignard, et Alexandre Denis. 2009. « MEDIA : a semantically annotated corpus of task oriented dialogs in French ». *Language Resources and Evaluation*, vol. 43, n° 4, p. 329–354.
- Bourguet, Marie-Luce. 2003. « Designing and Prototyping Multimodal Commands. ». In *INTERACT*. p. 717–720. Citeseer.
- Busemann, Stephan, Stephan Oepen, Elizabeth A. Hinkelman, Günter Neumann, et Hans Uszkoreit. 2011. « COSMA-multi-participant NL interaction for appointment scheduling ».
- Chris Messina. 2015. « Commerce Conversationnel », Blog. <<https://medium.com/chris-messina/conversational-commerce-92e0bccfc3ff#.vcxmj6jbs>>.
- Colby, Kenneth Mark. 1981. « Modeling a paranoid mind ». *Behavioral and Brain Sciences*, vol. 4, n° 04, p. 515–534.
- da Costa, Thiago Martini, Paulo Lísias Salomão, Amilton Souza Martha, Ivan Torres Pisa, et Daniel Sigulem. 2010. « The impact of short message service text messages sent as appointment reminders to patients' cell phones at outpatient clinics in São Paulo, Brazil ». *International Journal of Medical Informatics*, vol. 79, n° 1, p. 65–70.

- De Mori, R., F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, et G. Tur. 2008. « Spoken language understanding ». *IEEE Signal Processing Magazine*, vol. 25, n° 3, p. 50–58.
- Di Fabbri, Giuseppe, Dawn Dutton, Narendra K Gupta, Barbara Hollister, Mazin G Rahim, Giuseppe Riccardi, Robert E Schapire, et Juergen Schroeter. 2002. « AT&t help desk. ». In *INTERSPEECH*.
- Downer, Sean R, John G Meara, et Annette C Da Costa. 2005. « Use of SMS text messaging to improve outpatient attendance ». *Medical Journal of Australia*, vol. 183, n° 7, p. 366.
- El Asri, Layla, Romain Laroche, et Olivier Pietquin. 2014a. « DINASTI : Dialogues with a Negotiating Appointment Setting Interface. ». In *LREC*. p. 272–278.
- El Asri, Layla, Remi Lemonnier, Romain Laroche, Olivier Pietquin, et Hatim Khouzaimi. 2014b. « NASTIA : Negotiating Appointment Setting Interface. ». In *LREC*. p. 266–271.
- Fillmore, Charles J. 1985. « Frames and the semantics of understanding ». *Quaderni di semantica*, vol. 6, n° 2, p. 222–254.
- Goddeau, David, Helen Meng, Joseph Polifroni, Stephanie Seneff, et Senis Busayapongchai. 1996. « A form-based dialogue manager for spoken language applications ». In *Fourth International Conference on Spoken Language (ICSLP, 1996)*. (Philadelphia 1996), p. 701–704. IEEE.
- Godfrey, John J., Edward C. Holliman, et Jane McDaniel. 1992. « SWITCHBOARD : Telephone speech corpus for research and development ». In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP, 1992)*. (San Francisco 1992), p. 517–520. IEEE.
- Gorin, Allen L., Giuseppe Riccardi, et Jeremy H. Wright. 1997. « How may I help you ? ». *Speech communication*, vol. 23, n° 1, p. 113–127.
- Griol, David, Lluís F. Hurtado, Encarna Segarra, et Emilio Sanchis. 2008. « A statistical approach to spoken dialog systems design and evaluation ». *Speech Communication*, vol. 50, n° 8–9, p. 666 – 682.
- Hauswald, Johann, Michael A Laurenzano, Yunqi Zhang, Cheng Li, Austin Rovinski, Arjun Khurana, Ronald G Dreslinski, Trevor Mudge, Vinicius Petrucci, Lingjia Tang, et al. 2015. « Sirius : An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers ». In *ACM SIGPLAN Notices*. p. 223–238. ACM.
- He, Yulan et Steve Young. 2003. « Hidden vector state model for hierarchical semantic parsing ». In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*. (Hong Kong 2003), p. I–268. IEEE.

- Hemphill, Charles T., John J. Godfrey, et George R. Doddington. 1990. « The ATIS spoken language systems pilot corpus ». In *Proceedings of the DARPA speech and natural language workshop*. p. 96–101.
- journal du Net, Le. 2015. « Le nombre mondial d’abonnés au téléphone », En ligne. <<http://www.journaldunet.com/ebusiness/internet-mobile/1009553-monde-le-nombre-d-abonnes-au-telephone-mobile/>>.
- Jurafsky, Dan et James H Martin, 2014. *Speech and language processing*. Pearson.
- Koshy, Elizabeth, Josip Car, et Azeem Majeed. 2008. « Effectiveness of mobile-phone short message service (SMS) reminders for ophthalmology outpatient appointments : Observational study ». *BMC Ophthalmology*, vol. 8, n° 1, p. 9.
- Kuhn, Roland et Renato De Mori. 1995. « The application of semantic classification trees to natural language understanding ». *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, n° 5, p. 449–460.
- Larsson, Staffan et David R Traum. 2000. « Information state and dialogue management in the TRINDI dialogue move engine toolkit ». *Natural language engineering*, vol. 6, n° 3&4, p. 323–340.
- Lcock, Graham WI. 2012. « WikiTalk : A spoken Wikipedia-based open-domain knowledge access system ». In *24th International Conference on Computational Linguistics*. (Bombay 2012), p. 57.
- Lemeunier, Thierry. 2000. « L’intentionnalité communicative dans le dialogue homme-machine en langue naturelle ». PhD thesis, Université du Maine.
- Lemon, Oliver. 2011. « Learning what to say and how to say it : Joint optimisation of spoken dialogue management and natural language generation ». *Computer Speech & Language*, vol. 25, n° 2, p. 210–221.
- Letzte Änderung. 2015. « HeidelTime ». <<http://dbs.ifi.uni-heidelberg.de/index.php?id=129>>.
- Levin, Esther, Roberto Pieraccini, et Wieland Eckert. 2000. « A stochastic model of human-machine interaction for learning dialog strategies ». *IEEE Transactions on speech and audio processing*, vol. 8, n° 1, p. 11–23.
- Lison, Pierre et Raveesh Meena. 2014. « Spoken Dialogue Systems : The New Frontier in Human-computer Interaction ». *XRDS*, vol. 21, n° 1, p. 46–51.
- Litman, Diane J. et Scott Silliman. 2004. « ITSPOKE : An intelligent tutoring spoken dialogue system ». In *Demonstration papers at HLT-NAACL 2004*. p. 5–8. Association for Computational Linguistics.
- M. Smart, John. 2016. « The Conversational Interface and Your Personal Sim : Our Next Great Leap Forward », blog. <<http://www.accelerationwatch.com/lui.html>>.

- McTear, Michael F. 2002. « Spoken dialogue technology : enabling the conversational user interface ». *ACM Computing Surveys (CSUR)*, vol. 34, n° 1, p. 90–169.
- Meurs, Marie-Jean. 2009. « Approche stochastique bayésienne de la composition sémantique pour les modules de compréhension automatique de la parole dans les systèmes de dialogue homme-machine ». PhD thesis, RWTH, Aachen.
- Meurs, Marie-Jean, Fabrice Lefevre, et Renato De Mori. 2009. « Spoken language interpretation : On the use of dynamic bayesian networks for semantic composition ». In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. (Taipei 2009), p. 4773–4776. IEEE.
- Miller, Scott, Robert Bobrow, Robert Ingria, et Richard Schwartz. 1994. « Hidden understanding models of natural language ». In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. p. 25–32. Association for Computational Linguistics.
- Mohsin, Mohamad, Roslina Mohd Sidek, et A Noraziah. 2011. « WAS-GN : Web-based appointment system with GSM network ». *International Journal of Advancements in Computing Technology*, vol. 3, p. 315–333.
- Moriceau, Véronique et Xavier Tannier. 2013. « French resources for extraction and normalization of temporal expressions with heideltime ». *extraction*, vol. 26.
- Nio, Lasguido, Sakriani Sakti, Graham Neubig, Tomoki Toda, Mirna Adriani, et Satoshi Nakamura. 2014. Developing non-goal dialog system based on examples of drama television. *Natural Interaction with Robots, Knowbots and Smartphones*, p. 355–361. Springer.
- Papineni, Kishore, Salim Roukos, et Todd Ward. 1999. « Free-flow dialog management using forms. ». In *EUROSPEECH*.
- Pieraccini, Roberto et Juan Huerta. 2005. « Where do we go from here ? Research and commercial spoken dialog systems ». In *6th SIGdial Workshop on Discourse and Dialogue*.
- Pieraccini, Roberto, Esther Levin, et Chin-Hui Lee. 1991. « Stochastic Representation of Conceptual Structure in the ATIS Task. ». In *HLT*.
- Pustejovsky, James, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et others. 2003. « The timebank corpus ». In *Corpus linguistics*. p. 40.
- Pustejovsky, James, Kiyong Lee, Harry Bunt, et Laurent Romary. 2010. « ISO-TimeML : An International Standard for Semantic Annotation. ». In *LREC*.
- Rambow, Owen, Srinivas Bangalore, et Marilyn Walker. 2001. « Natural language generation in dialog systems ». In *Proceedings of the first international conference on Human language technology research*. p. 1–4. Association for Computational Linguistics.

- Ratté, Sylvie. 2015. « MTI830 : Forage de textes et de données audiovisuelle : L'art de forer et d'analyser des textes ». Programme de maîtrise en Génie des Technologies de l'Information. Montréal : Ecole de technologie supérieur.
- Raux, Antoine et Maxine Eskenazi. 2012. « Optimizing the turn-taking behavior of task-oriented spoken dialog systems ». *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 9, n° 1, p. 1.
- Reiter, Ehud. 1994. « Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible ? ». In *Proceedings of the Seventh International Workshop on Natural Language Generation*. p. 163–170. Association for Computational Linguistics.
- Ren, Fujii, Yu Wang, et Changqin Quan. 2015. « TFSM-based dialogue management model framework for affective dialogue systems ». *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 10, n° 4, p. 404–410.
- Ribeiro, Eugénio, Ricardo Ribeiro, et David Martins de Matos. 2015. « The Influence of Context on Dialogue Act Recognition ». *arXiv preprint arXiv :1506.00839*.
- Rieser, Verena et Oliver Lemon, 2011. *Reinforcement learning for adaptive dialogue systems : a data-driven methodology for dialogue management and natural language generation*. Springer Science & Business Media.
- Sacks, Harvey, Emanuel A Schegloff, et Gail Jefferson. 1974. « A simplest systematics for the organization of turn-taking for conversation ». *language*, p. 696–735.
- Schmid, Helmut. 1995. « Treetaggerl a language independent part-of-speech tagger ». *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, p. 28.
- Searle, John R, 1969. *Speech acts : An essay in the philosophy of language*, volume 626. Cambridge university press.
- Seneff, Stephanie. 1992. « TINA : A natural language system for spoken language applications ». *Computational linguistics*, vol. 18, n° 1, p. 61–86.
- Seneff, Stephanie et Joseph Polifroni. 2000. « Dialogue management in the Mercury flight reservation system ». In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*. p. 11–16. Association for Computational Linguistics.
- Skantze, Gabriel et Samer Al Moubayed. 2012. « IrisTK : a statechart-based toolkit for multi-party face-to-face interaction ». In *Proceedings of the 14th ACM international conference on Multimodal interaction*. p. 69–76. ACM.
- SRI. 2016. « Cognitive Assistant that Learns and Organizes ». <<http://www.ai.sri.com/project/CALO>>.

- Stent, Amanda, Rashmi Prasad, et Marilyn Walker. 2004. « Trainable sentence planning for complex information presentation in spoken dialog systems ». In *Proceedings of the 42nd annual meeting on association for computational linguistics*. p. 79. Association for Computational Linguistics.
- Stolcke, Andreas, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, et Marie Meteer. 2000. « Dialogue act modeling for automatic tagging and recognition of conversational speech ». *Computational linguistics*, vol. 26, n° 3, p. 339–373.
- Stone, Matthew, Christine Doran, Bonnie Webber, Tonia Bleam, et Martha Palmer. 2003. « Microplanning with communicative intentions : The SPUD system ». *Computational Intelligence*, vol. 19, n° 4, p. 311–381.
- Stoyanchev, Svetlana. 2015. « Seminar on Spoken Dialogue Systems ». <<http://www.cs.columbia.edu/~sstoyanchev/SDS2015/>>.
- Strötgen, Jannik et Michael Gertz. 2010. « HeidelTime : High quality rule-based extraction and normalization of temporal expressions ». In *Proceedings of the 5th International Workshop on Semantic Evaluation*. p. 321–324. Association for Computational Linguistics.
- Strötgen, Jannik et Michael Gertz. 2012. « Temporal Tagging on Different Domains : Challenges, Strategies, and Gold Standards ». In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. (Istanbul, Turkey 2012), p. 3746–3753. ELRA.
- Suomi, Reima, Ari Serkkola, et Markku Mikkonen. 2007. « GSM-based SMS time reservation system for dental care ». *International Journal of Technology and Human Interaction*, vol. 3, n° 3, p. 54.
- Taylor, Nicholas F., Judy Bottrell, Katherine Lawler, et Deenika Benjamin. 2012. « Mobile Telephone Short Message Service Reminders Can Reduce Nonattendance in Physical Therapy Outpatient Clinics : A Randomized Controlled Trial ». *Archives of Physical Medicine and Rehabilitation*, vol. 93, n° 1, p. 21–26.
- The Open University. 2014. « Statistiques sur l'utilisation des SMS », site. <<http://www.openuniversity.edu/news/news/2014-text-messaging-usage-statistics#>>.
- Turing, A. M. 1950. « Computing Machinery and Intelligence ». *Mind*, vol. 59, n° 236, p. 433–460.
- Valenta, Tomáš, Jan Švec, et Luboš Šmídl. 2012. « Spoken dialogue system design in 3 weeks ». In *International Conference on Text, Speech and Dialogue*. p. 624–631. Springer.
- VoiceXML. 2016. « VoiceXML's History », En ligne. <<http://www.voicexml.org/tutorials-2/introduction/voicexmls-history/>>.

- Wallace, Richard. 2003. « Be your own botmaster ». *Alice AI Foundation Inc.*
- Weizenbaum, Joseph. 1966. « ELIZA—a computer program for the study of natural language communication between man and machine ». *Communications of the ACM*, vol. 9, n° 1, p. 36–45.
- World Wide Web Consortium. 2007. « Voice Extensible Markup Language (VoiceXML) 2.1 ». <<https://www.w3.org/TR/voicexml21/>>.
- Young, Steve, Milica Gasic, Blaise Thomson, et Jason D. Williams. 2013. « POMDP-Based Statistical Spoken Dialog Systems : A Review ». *Proceedings of the IEEE*, vol. 101, n° 5, p. 1160–1179.