THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *29/11/2013* **par :**

Anaïs CADILHAC

Preference Extraction and Reasoning in Negotiation Dialogues

JURY

Jonathan GINZBURG	Professeur, Université Paris Diderot	Rapporteur
Bruno ZANUTTINI	Maître de Conférence, UCBN Caen	Rapporteur
Nicholas ASHER	Directeur de Recherche CNRS, IRIT Toulouse	Directeur de Thèse
Farah BENAMARA	Maître de Conférence, UPS Toulouse	Co-Directrice de Thèse
Claudette CAYROL	Professeur, UPS Toulouse	Présidente du Jury
Béatrice DAILLE	Professeur, Université de Nantes	Membre du Jury
Jérôme LANG	Directeur de Recherche CNRS, LAMSADE Paris	Membre du Jury

---

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (IRIT)
Directeur(s) de Thèse :
Nicholas ASHER et *Farah BENAMARA*
Rapporteurs :
Jonathan GINZBURG et *Bruno ZANUTTINI*

Anaïs Cadilhac

Preference Extraction and Reasoning in Negotiation Dialogues

Abstract

Modelling user preferences is crucial in many real-life problems, ranging from individual and collective decision-making to strategic interactions between agents for example. But handling preferences is not easy. Since agents don't come with their preferences transparently given in advance, we have only two means to determine what they are if we wish to exploit them in reasoning: we can infer them from what an agent says or from his nonlinguistic actions. Preference acquisition from nonlinguistic actions has been wildly studied within the Artificial Intelligence community. However, to our knowledge, there has been little work that has so far investigated how preferences can be efficiently elicited from users using Natural Language Processing (NLP) techniques.

In this work, we propose a new approach to extract and reason on preferences expressed in negotiation dialogues. After having extracted the preferences expressed in each dialogue turn, we use the discursive structure to follow their evolution as the dialogue progresses. We use CP-nets, a model used for the representation of preferences, to formalize and reason about these extracted preferences. The method is first evaluated on different negotiation corpora for which we obtain promising results. We then apply the end-to-end method with principles from Game Theory to predict trades in the win-lose game *The Settlers of Catan*. Our method shows good results, beating baselines that don't adequately track or reason about preferences.

This work thus presents a new approach at the intersection of several research domains: Natural Language Processing (for the automatic preference extraction and the reasoning on their verbalisation), Artificial Intelligence (for the modelling and reasoning on the extracted preferences) and Game Theory (for strategic action prediction in a bargaining game).

Keywords: Preferences, Dialogues, CP-nets, Discursive structure, NLP.

Anaïs Cadilhac

Extraction et Raisonnement sur les préférences dans des dialogues de négociation

Résumé

Modéliser les préférences des utilisateurs est incontournable dans de nombreux problèmes de la vie courante, que ce soit pour la prise de décision individuelle ou collective ou le raisonnement stratégique par exemple. Cependant, il n'est pas facile de travailler avec les préférences. Comme les agents ne connaissent pas complètement leurs préférences à l'avance, nous avons seulement deux moyens de les déterminer pour pouvoir raisonner ensuite : nous pouvons les inférer soit de ce que les agents disent, soit de leurs actions non-linguistiques. Plusieurs méthodes ont été proposées en Intelligence Artificielle pour apprendre les préférences à partir d'actions non-linguistiques mais à notre connaissance très peu de travaux ont étudié comment éliciter efficacement les préférences verbalisées par les utilisateurs grâce à des méthodes de Traitement Automatique des Langues (TAL).

Dans ce travail, nous proposons une nouvelle approche pour extraire et raisonner sur les préférences exprimées dans des dialogues de négociation. Après avoir extrait les préférences de chaque tour de dialogue, nous utilisons la structure discursive pour suivre leur évolution au fur et à mesure de la conversation. Nous utilisons les CP-nets, un modèle de représentation des préférences, pour formaliser et raisonner sur ces préférences extraites. Cette méthode est d'abord évaluée sur différents corpus de négociation pour lesquels les résultats montrent que la méthode est prometteuse. Nous l'appliquons ensuite dans sa globalité avec des raisonnements issus de la Théorie des Jeux pour prédire les échanges effectués, ou non, dans le jeu de marchandage *Les Colons de Catane*. Les résultats obtenus montrent des prédictions significativement meilleures que celles de quatre baselines qui ne gèrent pas correctement le raisonnement stratégique.

Cette thèse présente donc une nouvelle approche à la croisée de plusieurs domaines : le Traitement Automatique des Langues (pour l'extraction automatique des préférences et le raisonnement sur leur verbalisation), l'Intelligence Artificielle (pour la modélisation et le raisonnement sur les préférences extraites) et la Théorie des Jeux (pour la prédiction des actions stratégiques dans un jeu de marchandage).

Mots-clés : Préférences, Dialogues, CP-nets, Structure discursive, TAL.

Remerciements

Merci à mes deux encadrants, Nicholas Asher et Farah Benamara, pour m'avoir accompagnée dans ces trois années de thèse. Leur présence, leur écoute, leur complémentarité et leur gentillesse m'ont aidée tout au long du parcours et permis de réaliser ce travail dans de très agréables conditions.

Merci à mes deux rapporteurs, Jonathan Ginzburg et Bruno Zanuttini, pour avoir accepté de relire ma thèse et de l'avoir fait avec autant d'attention. De par leurs expertises très complémentaires, leurs commentaires et leurs remarques m'ont permis d'améliorer ce manuscrit et d'enrichir les perspectives de ce travail.

Merci également à Claudette Cayrol, Béatrice Daille et Jérôme Lang pour avoir accepté d'être membre de mon jury. Là encore, leurs profils très différents ont contribué à des questions et des discussions qui m'ont permis d'élargir ma vision de ce travail et m'ont ouvert des pistes de recherche plus passionnantes les unes que les autres.

I would like to thank Alex Lascarides for her many comments, her kindness and continued support and for the fact that she was always ready to help during our many collaborations.

I also want to thank ERC grant 269427 (STAC), which supported my research via travel expenses for the presentation of my work in several conferences. And I want to thank all the members of this project whom it was nice meeting and working with.

Merci à tous les membres de l'équipe MELODI pour m'avoir offert un environnement de travail très convivial. Merci particulièrement à Nathalie Aussenac-Gilles pour m'avoir initiée à la recherche avec Farah pendant mon stage de fin d'études et merci aussi à elle pour sa bienveillance tout au long de ma thèse. Merci à Ollivier

Haemmerlé et Nathalie Hernandez qui ont éveillé mon goût à la recherche par leurs enseignements passionnants à l'INSA. Merci à Stergos Afantenos, Philippe Muller et tous les autres membres de l'équipe et plus généralement du thème 4 de l'IRIT pour leurs apports à ce travail mais aussi aux moments plus relaxants, notamment les pauses du midi.

I would like to thank the Centre for Language Technology (CLT) people who welcomed me during my three month stay at Sydney Macquarie University. And I have a special thanks for Diego Mollá and Abeed Sarker for their friendly collaboration.

Merci à toute l'équipe pédagogique de l'Université de Toulouse qui m'a permis d'effectuer des enseignements divers et variés au cours de ces 3 années de thèse dans de très bonnes conditions. Et merci à tous mes étudiants d'avoir rendu cette expérience si enrichissante et pleine de bons souvenirs.

Merci à Mohamadou Seck pour l'intérêt qu'il a porté à notre sujet de stage et pour l'enthousiasme et la volonté dont il a fait preuve pendant les quelques mois passés à nos côtés.

Merci à toute notre tribu de doctorants, stagiaires, postdocs, pour avoir rendu cette thèse si agréable ! Merci à ceux qui m'ont offert une place chaleureuse dans leur bureau, à ceux qui ont entretenu cette ambiance dans le bureau au fur et à mesure des années et à ceux des bureaux voisins pour l'avoir partagée sans tenir compte des murs. Merci à ceux qui m'ont fait prendre goût au monde de la recherche par les bons moments partagés du temps où j'étais stagiaire et les moments plus rares mais tout aussi précieux partagés pendant mes années de thèse. Merci à ceux qui était là dès le début dans l'équipe MELODI et aux jeunes doctorants qui sont arrivés plus tard et m'ont fait me sentir vieille en début de 3ème année ! Merci à ceux avec qui j'ai pu partager la découverte de l'enseignement, la préparation des TP et les récits de nos expériences. Et plus généralement, merci à tous ceux qui ont contribué à la bonne ambiance à l'IRIT mais aussi au-delà des murs du labo à travers ces années.
Merci donc à toute la tribu : Anke, Antoine et Antoine, Baptiste, Bilal, Camille, Cédric, Chloé, Damien, François, Jean-Philippe, Julien, Juliette, Louis-Pierre, Manu, Maria, Marwa, Nadine, Newton, Nico, Pablo, Pierre, Philippe, Sarah, Seif, Serdjan, Soumya, Vladimir et bien d'autres...
Et dans le lot, je tiens à remercier tout particulièrement ceux qui ont traversé la thèse du tout début à la toute fin avec moi : Nadine, sans qui le bureau 303 aurait

été bien moins chaleureux, Camille avec qui j'ai pu vivre les aventures MELODI, et surtout Pierre et Anke avec qui j'ai pu découvrir, partager et vivre sereinement toutes les étapes de la thèse :)

Et finalement, merci à tous ceux qui m'ont encouragée de près ou de loin dans cette thèse et qui ont contribué à la réussite de ce travail grâce à tous les précieux moments passés à leurs côtés. Je voudrais, sans les nommer, remercier très chaleureusement ma famille et mes amis qui ont tenu ce rôle à merveille. Merci particulièrement à mes parents et ma sœur pour leur soutien inconditionnel. Et merci à mon N'ours pour son réconfort et sa bonne humeur, solutions inégalables contre le stress.

Contents

Contents 11

List of Tables 17

List of Figures 19

Introduction 21

1 Background 25

1.1 What are preferences? . . . 26

1.1.1 A formal definition of preferences . . . 26

1.1.2 Preferences vs. opinions . . . 27

1.2 Preferences in Game theory . . . 28

1.3 Preferences in Artificial Intelligence . . . 30

1.3.1 Preference acquisition . . . 30

1.3.2 Preference representation and reasoning . . . 35

1.3.2.1 Languages of propositional logic . . . 35

1.3.2.2 Constraint satisfaction . . . 38

1.3.2.3 Graphical languages . . . 38

1.3.2.4 Our choice for preference modelling . . . 42

1.3.3 CP-net, a model for qualitative preferences . . . 42

1.4 Preferences in Natural Language Processing . . . 44

1.5 Conclusion . . . 45

2 Preference change **47**

2.1 Intrinsic and extrinsic preference changes 48

2.1.1 Extrinsic preferences aren't always calculable 49

2.1.2 Intrinsic preferences can change 54

2.1.2.1 Preference change due to the discovery of new state descriptions 55

2.1.2.2 Preference change in the absence of any belief change 57

2.1.3 Interim conclusion 58

2.2 CP-nets, a consistent model for preference change 59

2.3 Related work about preference change 65

2.3.1 Global decision models 66

2.3.1.1 Two examples of extrinsic preference change . . . 67

2.3.1.2 Two examples of intrinsic preference change . . . 72

2.3.2 Changing utilities or beliefs? 74

2.4 Conclusion 76

3 Data and preference annotation **77**

3.1 Presentation of each corpus genre 78

3.1.1 ***Verbmobil*** and ***Booking*** corpora: cooperative negotiations 78

3.1.1.1 Presentation of the dialogues 78

3.1.1.2 The discourse-level annotation 79

3.1.2 ***Settlers*** corpus: a competitive game 83

3.1.2.1 Presentation of the game 83

3.1.2.2 The strategic annotation 85

3.2 Preference annotation 89

3.2.1 How are preferences linguistically expressed? 89

3.2.2 Preference annotation scheme for cooperative dialogues (***Verbmobil*** and ***Booking*** corpora) 91

3.2.2.1 Preference formalisation 91

3.2.2.2 Inter-annotator agreements 92

3.2.3 Preference annotation scheme for non-cooperative dialogues (***Settlers*** corpus) . . . 99

3.2.3.1 How preferences are expressed in the ***Settlers*** corpus? . . . 100

3.2.3.2 Extension of the annotation scheme . . . 101

3.2.3.3 Inter-annotator agreements . . . 103

3.3 Conclusion . . . 107

4 Preference extraction **109**

4.1 Outcome extraction . . . 110

4.1.1 External resources . . . 110

4.1.1.1 Domain Ontology . . . 110

4.1.1.2 Lexicon . . . 111

4.1.2 Classifier and feature set . . . 112

4.1.2.1 Local features . . . 112

4.1.2.2 Discursive features . . . 113

4.1.3 Experiments and results . . . 114

4.2 Preference identification . . . 116

4.2.1 Outcome acceptance . . . 117

4.2.2 Outcome attachment . . . 118

4.2.3 Outcome dependencies . . . 119

4.3 Conclusion . . . 119

5 Discourse and preference evolution **121**

5.1 CP-net descriptions . . . 123

5.2 Rules to construct preferences from EDUs . . . 124

5.2.1 From preferences in EDU together with rhetorical relations to partial preference descriptions . . . 124

5.2.1.1 Commit . . . 125

5.2.1.2 IExplanation and Elaboration . . . 128

5.2.1.3 Question-answer pair . . . 132

5.2.1.4 Correction . . . 133

5.2.1.5 Other relations 134
5.2.2 From preference descriptions to models 134
5.2.3 Treatment of an example 135
5.3 Evaluation of the proposed method 139
5.4 Conclusion 142

6 Preferences and strategy prediction **143**
6.1 Dialogue act and resource prediction 144
6.1.1 Identifying dialogue acts 145
6.1.2 Finding resource text spans 146
6.1.3 Recognizing the type of resources 147
6.2 Predicting players' strategic actions 148
6.2.1 CP-Nets in our *Settlers* corpus 149
6.2.2 Modelling players' preferences 149
6.2.3 Example 151
6.2.4 Evaluation and results 153
6.3 Conclusion 156

Conclusion **159**

Appendices **165**

A Glossary **167**
A.1 The AGM model for belief revision 167
A.2 Evaluation measures for inter-annotator agreements 168
A.3 Evaluation measures for classification tasks 169
A.4 The McNemar's test 170

B Résumé en français **171**
Introduction 172
B.1 Préférences : état de l'art 173
B.1.1 Préférences en Théorie des jeux 174

B.1.2 Préférences en Intelligence Artificielle 174
B.1.2.1 Différents systèmes d'acquisition des préférences 175
B.1.2.2 Les CP-nets, une méthode de formalisation des préférences 176
B.1.3 Préférences dans le Traitement Automatique des Langues . 177
B.2 Corpus et annotation des préférences 179
B.2.1 Présentation des corpus 180
B.2.1.1 Les corpus ***Verbmobil*** et ***Booking*** : des négociations coopératives 180
B.2.1.2 Le corpus ***Settlers*** : un jeu compétitif 182
B.2.2 Comment les préférences sont-elles exprimées ? 184
B.2.3 Annotation des préférences 185
B.2.3.1 Annotation dans les corpus ***Verbmobil*** et ***Booking*** 185
B.2.3.2 Evolution du modèle d'annotation pour le corpus ***Settlers*** 186
B.2.3.3 Evaluation du modèle 187
B.3 Extraction automatique des préférences 187
B.3.1 Identification des options 188
B.3.2 Identification des préférences sur les options 192
B.4 Formaliser l'évolution des préférences 194
B.4.1 Les règles d'évolution des préférences 194
B.4.1.1 Des préférences dans chaque UDE à la description partielle des préférences 195
B.4.1.2 De la description des préférences au CP-net minimal associé 198
B.4.2 Traitement d'un exemple 199
B.4.3 Conclusion 201
B.5 Prédiction de stratégies 202
B.5.1 Pré-traitement : Prédiction des actes de dialogues et des préférences sur les ressources 202
B.5.1.1 Identification des des actes de dialogues 202
B.5.1.2 Identification des préférences sur les ressources . 203

B.5.2 Prédiction des actions stratégiques des joueurs 203
B.5.2.1 Prédire les échanges à partir des préférences . . . 204
B.5.2.2 Exemple 206
B.5.2.3 Evaluation et résultats 207
B.6 Conclusion 211

Bibliography **213**

List of Tables

3.1 The strategic annotation for a ***Settlers*** dialogue. 87
3.2 Statistics for the preference annotation in ***Verbmobil*** and ***Booking***. 92
3.3 Inter-annotator agreements for ***Verbmobil*** and ***Booking***. 93
3.4 Agreements on binary operators for ***Verbmobil*** and ***Booking***. . . . 96
3.5 Statistics for the preference annotation in ***Settlers***. 103
3.6 Inter-annotator agreements for the three corpora (***Verbmobil***, ***Booking*** and ***Settlers***). 104
3.7 Agreements on binary operators for the three corpora (***Verbmobil***, ***Booking*** and ***Settlers***). 106

4.1 Results for automatic preference extraction. 115
4.2 Results for automatic preference identification. 117
4.3 Results for outcome dependencies. 117

5.1 The discourse structure for a ***Verbmobil*** dialogue. 137
5.2 Evaluation for preference extraction from each EDU. 140
5.3 Evaluation for preference evolution through dialogue. 140

6.1 Results for dialogue act classification. 147
6.2 Results for resource type classification. 148
6.3 The negotiation annotation for a ***Settlers*** dialogue. 152
6.4 Results for trade prediction. 154
6.5 Results for the end to end trade prediction. 156

A.1 Confusion matrix. 169

B.1 Taux d'accord inter-annotateurs pour nos trois corpus. 187
B.2 Résultats (pourcentages) pour les trois évaluations. 190
B.3 Résultats pour la classification des actes de dialogues. 203
B.4 Résultats pour la classification du type de ressource. 204
B.5 L'annotation stratégique d'un dialogue du corpus ***Settlers***. 206
B.6 Résultats pour la prédiction des échanges. 209
B.7 Résultats pour la prédiction automatique globale des échanges. . . . 210

List of Figures

1.1 The movie recommender *MovieLens* interface. 31
1.2 The restaurant recommender *Entree* interface. 32
1.3 The apartment recommender *RentMe* interface. 32
1.4 The dynamic critiquing recommender interface. 33
1.5 The shopping recommender *ExpertClerk* interface. 34
1.6 A lexicographic preference tree for the travel example. 39
1.7 A CP-net illustration for the travel example. 40
1.8 A UCP-net illustration for the travel example. 41
1.9 A GAI-net illustration for the travel example. 42
1.10 CP-net and the induced preference graph for the travel example. . . 44

2.1 Backward induction algorithm. 51
2.2 Partial reasoning in big game. 54
2.3 The global decision models for EPC-a and EPC-b. 68
2.4 The global decision model for IPC-a and IPC-b. 73

3.1 The discourse structure for a ***Verbmobil*** dialogue. 82
3.2 *The Settlers of Catan* chat interface. 83
3.3 The discourse structure for a ***Settlers*** dialogue. 86

6.1 A CP-net example from the ***Settlers*** corpus. 149

B.1 Un exemple de CP-net pour l'exemple du vol avec escale. 177
B.2 Structure discursive d'un dialogue de ***Verbmobil***. 200
B.3 Un exemple de CP-net pour le corpus ***Settlers***. 205

Introduction

Preferences rule actions in our daily life ranging from choices of moderate importance like choosing a book to read or clothes to wear to more critical ones like choosing a job or a house to buy. Modelling user preferences is crucial in many real-life problems, ranging from individual and collective decision-making (Arora and Allenby, 1999) to strategic interactions between agents (Brainov, 2000) and game theory (Hausman, 2000). A web-based recommender system can, for example, help a user identify (among an optimal ranking) the product item that best fits his preferences (Burke, 2000).

But handling preferences is not easy. First, specifying an ordering over acceptable outcomes is not trivial especially when multiple aspects of an outcome matter (*outcome* refers to the objects of choice over which preference are expressed). For instance, choosing a new camera to buy may depend on several criteria (e.g. battery life, weight, etc.), hence, ordering even two outcomes (cameras) can be cognitively difficult because of the need to consider trade-offs and dependencies between the criteria. Secondly, users often lack complete information about preferences initially. They build a partial description of agents' preferences that typically changes over time. Indeed, users often learn about the domain, each others' preferences and even their own preferences during a decision-making process. Since agents don't come with their preferences transparently given in advance, we have only two means to determine what they are if we wish to exploit them in reasoning: we can infer them from what an agent says or from his nonlinguistic actions.

Preference acquisition from nonlinguistic actions has been wildly studied within the Artificial Intelligence community. Two main approaches have been proposed: *preference learning* (Fürnkranz and Hüllermeier, 2011) where preferences are acquired using a variety of methods including collaborative filtering and content-based recommender systems and *preference elicitation* (Chen and Pu, 2004) where the acquisition of preferences is the result of an interactive process with the user, generally by means of specific interfaces.

However, to our knowledge, there has been little work that has so far investigated how preferences can be efficiently elicited from users using Natural Language

Processing techniques. In this thesis, we propose a first step towards preference extraction in spontaneous conversation using a computational linguistic approach as well as a method that reason on these verbalised preferences. Our approach is at the intersection of Game Theory, Artificial Intelligence and Natural Language Processing and provides both theoretical and empirical contributions.

The first part of the thesis focuses on the theoretical contributions. In Chapter 1, we introduce some background about preferences. We first define the most important notions we will use throughout the dissertation. We then give an overview of previous work in the different research domains our work relates to. We also present a more detailed description of CP-net (Boutilier et al., 2004), the formalism we use to model and reason about preferences.

In Chapter 2, we propose a theoretical study of preference changes. Our analysis was motivated by state of the art assumption in most models of rational action that supposes that all possible states and actions are predefined and that preferences change only when beliefs do. In this chapter, we introduce several decision and game problems that lack these features, arguing that they call for a dynamic model of preferences: that is, preferences can change when unforeseen possibilities come to light or when there is no specifiable or measurable change in belief. We propose a formally precise dynamic model of preferences that extends and refines the existing static CP-net model of preferences. The axioms that update and revise preferences ensure that preferences remain consistent while minimising changes. We demonstrate that this dynamic model overcomes some of the problems observed in related work about preference change. This work is currently under submission in a journal (Cadilhac et al., submitted).

The second part of the manuscript presents our empirical contributions. It describes a computational linguistic approach for preference extraction and reasoning in negotiation and bargaining dialogues. In Chapter 3, we present our different corpora, study how preferences are linguistically expressed and propose a new scheme for their annotation. This work is performed on different corpus genres. First, we study two corpora where agents negotiate about how to carry out a common goal. For ***Verbmobil***, the goal is to find a meeting time, and for ***Booking*** it is to arrange a reservation. Then, we study the ***Settlers*** corpus, a corpus of on line chats concerning the non-cooperative bargaining game *The Settlers of Catan*. Negotiations in this game mirror complex real life negotiations and so provide a fruitful arena to study strategic conversation. We study how preferences are verbalised and show that agents' preferences depend upon the compositional interpretation of the discourse structure. We then describe the annotation methodology and detail the inter-annotator agreement study on each corpus genre. Our results show that preferences can be easily annotated by humans. This work has been published in two papers: first at the joint conference on Lexical and Computational Semantics (*SEM) (Cadilhac et al., 2012b) for the annotation of ***Verbmobil*** and ***Booking***, then at the Linguistic Annotation Workshop (LAW) (Cadilhac et al., 2012a) for ***Settlers***.

Based on the previous linguistic study, Chapter 4 explains how to automatically extract the preferences and their dependencies within each discourse unit. We propose a Natural Language Processing-based approach to extract the preferences from our corpora of negotiation dialogues. We perform the extraction in two steps: first, we extract the set of outcomes; then, we identify how these outcomes are ordered. For the first step, we use supervised learning with a combination of both local and discursive features, while the second step relies on a hybrid approach. We finally assess the reliability of our method on the ***Verbmobil*** and ***Booking*** corpora. This work has been published at the French joint conference JEP-TALN-RECITAL about Natural Language Processing (Cadilhac et al., 2012d) and at the European Conference on Artificial Intelligence (ECAI) (Cadilhac et al., 2012c).

In Chapter 5, we present general rules for translating the preferences from each discourse unit into an evolutionary description of CP-nets. This work complements the study of preference change in general presented in Chapter 2 since it provides more precise rules to model changes as they are verbalised by the agents. We extract constraints on preferences and dependencies among them, even when they are expressed indirectly, by exploiting discourse structure. The agents' preferences depend upon the compositional interpretation of the discourse structure and the constraints are different for different discourse relations, reflecting the fact that the semantics of connections between discourse units influences how their preferences relate to one another. Based on these discourse relations, our method gives a formal description of each agent's preferences at any moment in the dialogue and models the evolution of these preferences as the dialogue progresses. We test the algorithms predictions against the judgements of naive annotators on unseen dialogues. The average annotator-algorithm agreement and the average inter-annotator agreement show that our method is reliable. This work has been published at the Special Interest Group on Discourse and Dialogue (SIGdial) (Cadilhac et al., 2011).

Chapter 6 combines the two previous works (the automatic extraction of preference from Chapter 4 and the modelling of the preference evolution during the dialogue from Chapter 5) to predict trades in the win-lose game *The Settlers of Catan*. We exploit the conversation to dynamically construct a partial model of each player's preferences in dialogues from our ***Settlers*** corpus, which in turn yields equilibrium trading moves via principles from game theory. A comparison of our method against four baselines shows that tracking how preferences evolve through the dialogue and reasoning about equilibrium moves are both crucial to success.

This work has been published at the conference on Empirical Methods in Natural Language Processing (EMNLP) (Cadilhac et al., 2013).

Eventually, in the Conclusion, we provide an overview of this work and emphasise its progresses and limitations. We expose our perspectives for future work with a quick presentation of a work in progress with Diego Mollá and Abeed Sarker from Macquarie University in Sydney, which studies preferences in medical data, a completely different field.

Chapter 1

Background

1.1 What are preferences? **26**
1.1.1 A formal definition of preferences 26
1.1.2 Preferences vs. opinions 27
1.2 Preferences in Game theory **28**
1.3 Preferences in Artificial Intelligence **30**
1.3.1 Preference acquisition 30
1.3.2 Preference representation and reasoning 35
1.3.3 CP-net, a model for qualitative preferences 42
1.4 Preferences in Natural Language Processing **44**
1.5 Conclusion **45**

In this chapter, we introduce some background about preferences. First, in Section 1.1, we define the notion of preferences and the terms associated with it. We then distinguish preferences from opinions. Since they both imply some personal judgement towards objects or actions given by an agent, it is important to recall their differences and give some examples to illustrate the cases where they are really distinct from those where they get closer.

Having given these definitions, we continue our description of preferences by a review of previous work in different domain of research. We are interested in *Game Theory*, one of the first fields in which researchers became interested in the study of preferences (see Section 1.2); *Artificial Intelligence* which more recently has seen a lot of work about preference in both preference acquisition and preference reasoning (see Section 1.3); and *Natural Language Processing* which considers preferences as comparative opinions (see Section 1.4).

1.1 What are preferences?

A *preference* is commonly understood as an ordering by an agent over *outcomes*, which are understood as actions that the agent can perform or goal states that are the direct result of an action of the agent. For instance, an agent's preferences may be defined over actions like *buy a new car* or by its end result like *have a new car*. Outcomes can include states that nature or other agents control. For example, after buying a lottery ticket an agent may prefer to *win* rather than *lose* but he cannot control this. The outcomes over which a preference is defined will depend on the domain or task. In a web-based recommender system, they can concern various objects like cameras, cars, hotel rooms, flights. In an information retrieval system, they can concern several articles of which the user wants to retrieve the most appropriate. During a bargaining process, they can concern the traded products, the person with whom the trade is performed, etc.

Among these outcomes, some are *acceptable* for the agent, i.e. the agent is ready to act in such a way as to realize them, and some outcomes are not. Among the acceptable outcomes, the agent will typically prefer some to others. Our aim is not to determine the most preferred outcome of an agent but rather to follow the evolution of their commitments to certain preferences as the dialogue proceeds. To give an example, if an agent proposes to meet on a certain day X and at a certain time Y, we learn that among the agent's acceptable outcomes is a meeting on X at Y, even if this is not his most preferred outcome.

1.1.1 A formal definition of preferences

More formally, we note $\succeq$, a *preference relation* over elements of Ω, a set of possible outcomes. Given the two outcomes o_1 and o_2, $o_1 \succeq o_2$ means that outcome o_1 is equally or more preferred to the decision maker than o_2. The associated *indifference relation* where o_1 and o_2 are equally preferred is $o_1 \sim o_2$ if and only if $o_1 \succeq o_2$ and $o_2 \succeq o_1$. The associated *strict preference relation* $o_1 \succ o_2$ holds if and only if $o_1 \succeq o_2$ and not $o_2 \succeq o_1$.

The preference relation $\succeq$ is a reflexive ($o_1 \succeq o_1$ for every $o_1 \in \Omega$) and transitive ($o_1 \succeq o_3$ whenever $o_1 \succeq o_2$ and $o_2 \succeq o_3$ for $o_1, o_2, o_3 \in \Omega$) binary relation over elements of Ω. The indifference relation is symmetric ($o_2 \sim o_1$ whenever $o_1 \sim o_2$) while the strict preference relation is not ($\neg(o_2 \succ o_1)$ whenever $o_1 \succ o_2$).

The satisfaction of axioms on preferences leads us to define a notion of consistency for preferences: a preference ordering P is consistent if and only if we cannot deduce from P and the axioms that $a \succ b$ and $b \succ a$ for any a and b (or by transitivity that $a \succ a$ for any a) (Grüne-Yanoff and Hansson, 2009a).

The preferences are commonly taken to be complete orderings over the set of outcomes. The relation is complete if $o_1 \succeq o_2$ or $o_2 \succeq o_1$ or equivalently if $o_1 \succ o_2$ or $o_2 \sim o_1$ or $o_2 \succ o_1$ for every $o_1 \in \Omega$ and $o_2 \in \Omega$. If some candidates are not comparable by a given agent, the preference relation is incomplete with respect to them and the two alternatives are called "incomparable" (Hansson and Grüne-Yanoff, 2006).

As we are interested in how agents reveal their own preferences when they talk, we are interested in an *ordinal* (also called *qualitative*) definition of preferences. Indeed, this linguistic information almost always provides an ordinal definition of preferences, which consists in imposing a ranking over relevant possible outcomes and not a *cardinal* (or *quantitative*) definition based on numerical values that allow comparisons but which are not the natural way we express preferences.

1.1.2 Preferences vs. opinions

In Natural Language Processing, preferences are studied within the field of opinion mining (see Section 1.4), though preferences and opinions are distinct.

While opinions are defined as a point of view, a belief, a sentiment or a judgement that *an agent may have about an object or a person*, preferences, as we have defined them, involve an ordering on behalf of an agent and thus are *relational and comparative*. Hence, opinions concern absolute judgements towards objects or persons (positive, negative or neutral), while preferences concern relative judgements towards actions (preferring them or not over others). The following examples illustrate this:

(a) The scenario of the first season is not bad.

(b) The scenario of the first season is better than the second one.

(c) I prefer to watch the first season rather than the second one.

(a) expresses a direct positive opinion towards the scenario of the first season but we do not know if it is the most preferred. (b) expresses a comparative opinion between two seasons with respect to their shared features (scenarios) (Jindal and Liu, 2006b; Ganapathibhotla and Liu, 2008). If actions involving these seasons (e.g. seeing them) are clear in the context as in (c), such a comparative opinion will imply a preference, ordering the first season over the second.

Reasoning about preferences is also distinct from reasoning about opinions. An agent's preferences determine an order over outcomes that predicts how the agent, if he is rational, will act. This is not true for opinions. Opinions have at best an indirect link to action (we do what we like to do when possible), but preferences are an intrinsic part of the analysis of what it is to act rationally: I may hate what I'm doing, but do it anyway because I rationally prefer that outcome to any of the alternatives.

1.2 Preferences in Game theory

Preferences have been studied in economics, especially in decision theory, social choice and game theory since around the 1950s, long before Artificial Intelligence researchers became interested in the topic in the 1990s (Domshlak et al., 2011). In this section, we present preferences in the game theory context as it will be particularly useful to study preferences in the ***Settlers*** corpus (see Section 3.1.2), a corpus of on line chats concerning the competitive win-lose game *The Settlers of Catan*.

Game theory studies interactions between decision-makers. In traditional game theory, preferences or utilities over outcomes drive *rational*, *strategic* decision (opinions play no such role). A decision is *strategic* when each player cares not only about his own action but also about the actions taken by the other players. A decision-maker is *rational* when he is aware of his alternatives, forms expectations about any unknowns, has clear preferences, and chooses a feasible action deliberately after some process of optimization (Osborne and Rubinstein, 1994).

Outcomes in standard game theory are the set of possible actions from which the decision-maker makes a choice, or else the set of the possible *consequences* of these actions. The preference relation is a complete transitive reflexive binary relation as defined is Section 1.1. It is also common to define the preference model by giving a utility function u that attaches a number to each outcome: $o_1 \succeq o_2$ if and only if $u(o_1) \geq u(o_2)$ and $o_1 \sim o_2$ if and only if $u(o_1) = u(o_2)$.

What we ordinarily call preferences are captured in decision theory and game theory via *expected utility*, which is defined in terms of the agent's *beliefs* and *utility function*. To formalize a decision problem, we can use Savage's analytical framework (1954) consisting of:

- a set S of states of the world,
- an arbitrary set C of possible consequences,
- a set F of acts, that is, functions taking each state of the world to a consequence,

- and a preference relation over the acts on F which determines the agent's choice.

Both *beliefs* and *utility function* determine the agent's preferences over acts represented by his *expected utility*.

- The *beliefs* concerns the set S of states of the world. The agent's *beliefs* are represented by a *probability function* on the set of states over which the agent has no power.

- The *utility function* concerns the set C of consequences. The *attractiveness* of the consequences are represented by a real-valued *utility function*: the more attractive consequences have higher utility.

For a *probability function* p and a *utility function* u, the *expected utility* of an act $f \in F$ is $\sum_{s \in S} p(s).u(f(s))$. So for rational agents, the agent's preferred actions are the ones that maximise his expected utilities and are an optimal trade off between what the agent would prefer to achieve and what he thinks he can achieve.

Game theory postulates that agents calculate their actions based on a common knowledge of all the players' preferences. Solving a game problem involves finding *equilibrium strategies*: an optimal action for each player in that it maximises his expected utility, assuming that the other players perform their specified action (Shoham and Leyton-Brown, 2009). Calculating equilibria thus requires knowledge of the other players' preferences.

But in real life, strategic interactions almost always occur under the handicap of various forms of imperfect information. People don't know what other relevant actors are going to do, first because they typically don't know what they believe and what they want. In fact, we think that people often don't have knowledge of their own preferences or even what actions they can perform in a given situation; for instance, people compare shops to determine which goods of a certain type they might buy. A further complication is that in real life strategic situations can involve a great many possible actions; the underlying game is so large that agents with limited computational power can't hope to compute in analytical fashion the optimal actions they should perform. Instead, such agents strategize in subregions of the entire game. To do so, they must assign values or utilities to *intermediate* states of the game, whose values or utility may not always reflect the value of the end state (the valuation is not monotonic along initial segments of a complete strategy). Because a knowledge of preferences is crucial to informed strategic action, people try to extract information about the preferences of other agents and often provide information about their own preferences when they talk. We will detail this further in Chapter 2.

1.3 Preferences in Artificial Intelligence

Working with preferences involves three subtasks (Brafman and Domshlak, 2009; Kaci, 2011): *acquiring* the users' preferences, *modelling and representing* the preference information and *reasoning* in order to compute an answer to common queries given the model, like finding the optimal outcome, ordering the set of outcomes or aggregating the preferences of multiple users.

In this section, we start by a presentation of these subtasks and finish by a more detailed description of CP-nets (Boutilier et al., 2004), a compact model for representing and reasoning with preferences that we use in this work. As we will see in the next chapters, CP-nets can be successfully used to model and reason about changing preferences as they can be linguistically expressed by the agents.

1.3.1 Preference acquisition

Two main methods are used within the Artificial Intelligence community to acquire preferences (Kaci, 2011): *preference learning* where the system has to learn from data describing the user's behaviour or past preferences in order to make predictions about unseen preferences and *preference elicitation* where preferences are the result of an interactive process with the user. Preference learning is an "implicit" approach in the sense that the user is not actively involved in the acquisition task while preference elicitation is an "explicit" approach that requires specific preference input from the user (Pommeranz et al., 2012).

Preferences learning (Pazzani, 1999; de Gemmis et al., 2009; Fürnkranz and Hüllermeier, 2011) is used in recommender systems like *collaborative filtering* ones which recommend unseen items to an user based on the choices of users with similar profiles who have already rated these items (Resnick et al., 1994). *Content-based filtering* recommender systems try to predict the preferences of an user according to his own previous choices. Recommendations are made based on the similarities between the description of the items that have been rated by the user and the description of new items to be recommended (Pazzani and Billsus, 1997). *Demographic-Based* recommender systems exploit information like the user's age, gender or employment to predict the types of users that like an item (Krulwich, 1997). Other recommender systems exploit *knowledge* about outcomes and user needs in a particular domain and *hybrid* systems, at last, combine different recommendation techniques.

Several systems exploit different kinds of interactions with the user in order to elicit his preferences by answering some questions, using graphical interfaces to express some specific constraints or compare different outcomes (Chen and Pu, 2004; Pu and Chen, 2008; Pommeranz et al., 2012). Preference elicitation is used in recommender systems to help users to find the best outcome in a huge set of possible outcomes, typically through item-rating or more conversational interactions based on knowledge-based similarity or example critiquing.

An example of item-rating interface is presented in Figure 1.1[1]. The interface corresponds to the *MovieLens* recommender system where user can use pulldown interface to evaluate movies (Miller et al., 2003).

Figure 1.1: The movie recommender *MovieLens* interface.

The *FindMe* recommender systems (including *PickAFlick* for movies, *Entree* for restaurant, *RentMe* for apartment-finding) (Burke, 2000) exploit similarity retrieval: the user selects a given item from the catalog and requests other items similar to it. FindMe systems also exploit tweaks: the finding is also based on similarity but the resulting set of items is filtered to only keep items satisfying the user's tweak like "nicer" or "cheaper" (see Figures 1.2 and 1.3 for an illustration).

Figure 1.2: The restaurant recommender *Entree* interface.

Figure 1.3: The apartment recommender *RentMe* interface.

Chen and Pu (2007) present a critiquing-based recommender interface where users can generate critiques over single or multiple features, e.g. for digital camera "Different Manufacture, Lower Resolution and Cheaper" (see Figure 1.4 for an illustration).

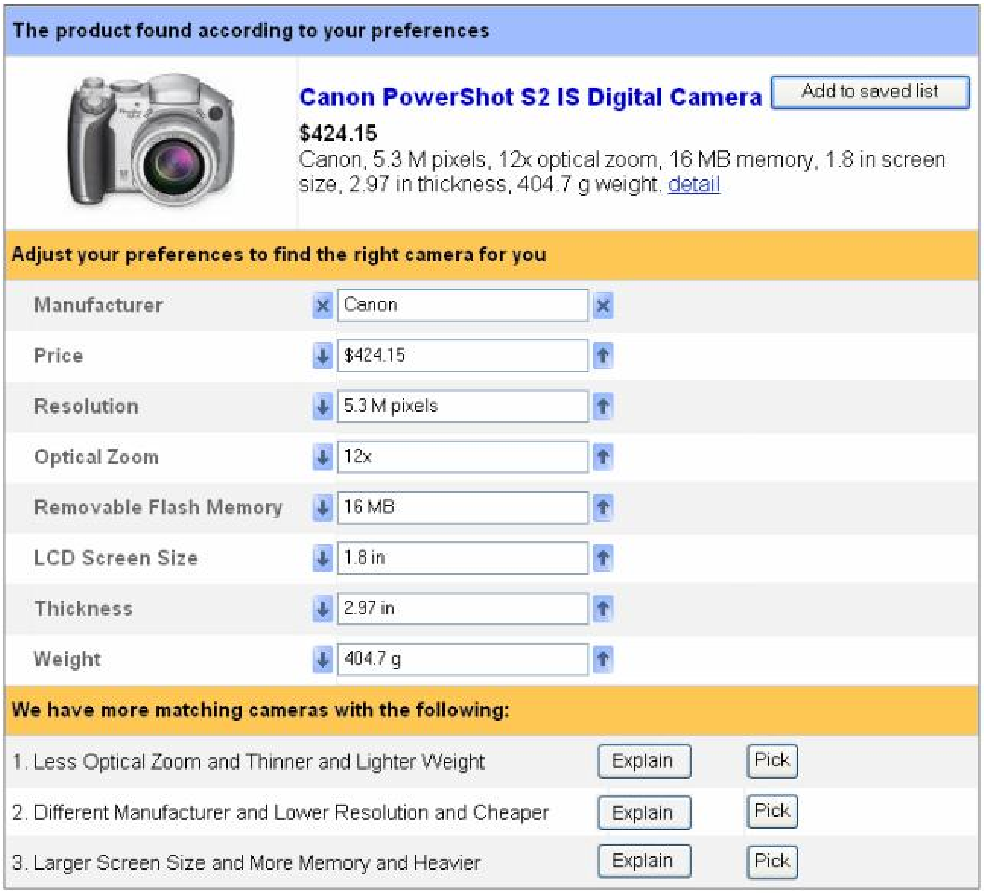


Figure 1.4: The dynamic critiquing recommender interface.

In the *ExpertClerk* system (Shimazu, 2001), a character agent talks with a shopper in a natural language to help him find, compare, and choose among a lot of merchandise goods (see Figure 1.5). The system elicits the customer's preferences in

[1]From http://www.movielens.org/html/tour/movies.html.

two ways: first, by asking effective questions, then by proposing three contrasting sample goods with explanations of their selling points and observing the customer's responses. For interacting with the user, the dialogue system uses the system from Shimazu et al. (1992) which provide a natural language interface to databases. It translates a user's request into a corresponding SQL query and issues the query to a backward relational merchandise database. The translation is performed by recognizing keywords and linguistic patterns associated with the database field names and values.

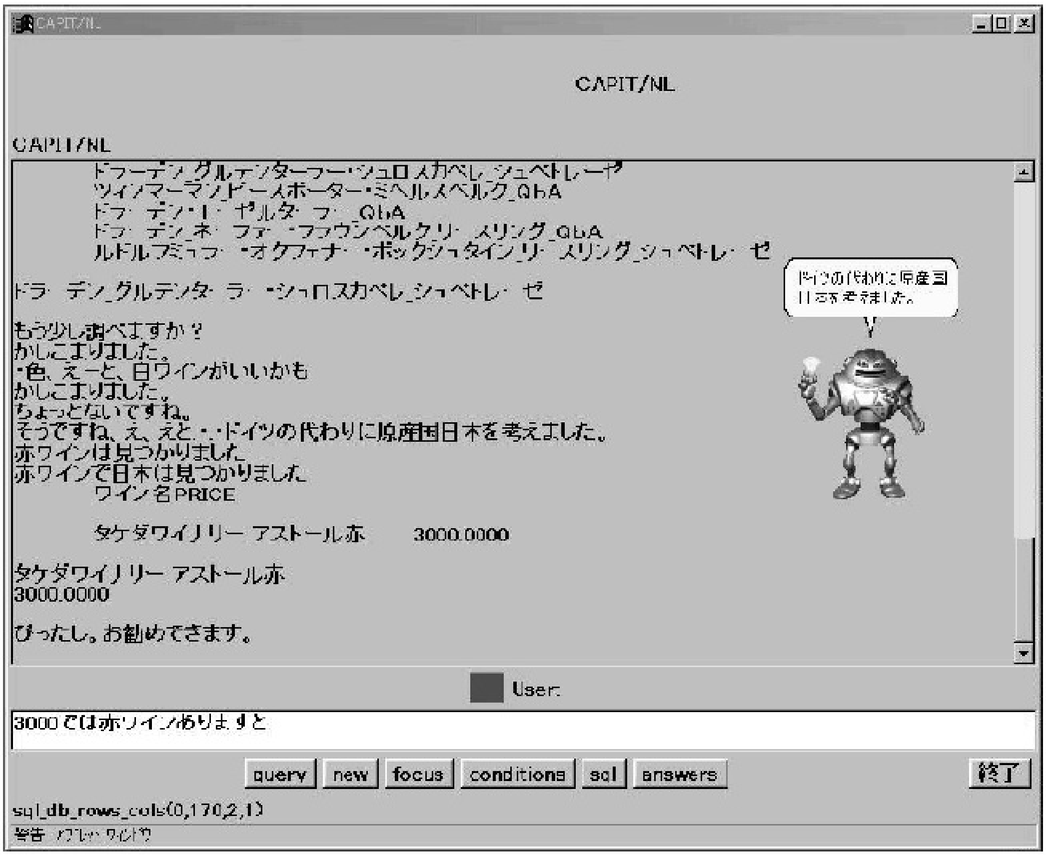


Figure 1.5: The shopping recommender *ExpertClerk* interface.

Decision support systems (Aloysius et al., 2006) elicit preferences to help users to take decisions typically more critical than buying items, e.g. in the medical domain. In order to have a precise model of the users preferences, the majority of decision support systems represent preferences in form of utility functions (see

Section 1.2). Two commonly used preference elicitation techniques in multi-criteria decision support systems are absolute measurement (each attribute is independently associated with a score in an absolute scale, e.g., I would rate annual salary as rating a score of 8 in importance on a scale from 1 to 10 in my job search, while distance driven to work would rate a score of 6) and pairwise comparison (attributes are compared in pairs to judge their relative importance, e.g., I would rate annual salary as being more important to my job search decision than the distance I drive daily to work). In elicitation interface, scores are typically entered on discrete scales by selecting a rating from a drop down list or using horizontally aligned radio buttons and on continuous scales by using a slider (Pommeranz et al., 2012).

1.3.2 Preference representation and reasoning

Several formalisms exist to represent preferences with different kind of approach (graphical, logic, etc.). However, most of them are closely related and preferences expressed in one formalism can be expressed in another either equivalently or with a good approximation (Coste-Marquis et al., 2004; Dubois et al., 2006; Domshlak et al., 2011).

1.3.2.1 Languages of propositional logic

Languages of propositional logic are one of the possible approach to represent preferences (Coste-Marquis et al., 2004; Domshlak et al., 2011).

Among these frameworks, we can cite models based on ***penalties*** (Dupin de Saint Cyr-Bannay et al., 1994) where the agent expresses his preferences in terms of goals, represented as propositional formulas, associated with weights, usually positive numbers which give the penalties associated with the non-satisfaction of the corresponding goal (so a greater penalty corresponds to a more important goal). The preferred alternative is the one with the lower global penalty calculated by summing the elementary penalties of each violated goal. In this framework, the weights are clearly significant in a quantitative way (with a numerical meaning and not only a ranking one) since penalties are compensatory ("the violation of a goal may be compensated by the satisfaction of a sufficient number of goals of lower importance").

Consider the following simple example 1.1 to illustrate this formalism.

(1.1) Suppose an agent prefers to go from Paris to Hong Kong by day rather than overnight. If he takes an overnight trip, he prefers a nonstop flight, but if he goes by day he prefers a flight with a stop.

Let, D be the variable for the preference over the period of travel and S the variable for the preference over stops. The domain of D is $\{day, \overline{day}\}$, where day is a day trip and $\overline{day}$ is a night one. The domain of S is $\{stop, \overline{stop}\}$ where $stop$ is a trip with stops and $\overline{stop}$ is one without.

The preferences of the agent can be expressed with the following goal base GB where each goal is a propositional formula with an associated weight.

$$GB = \left\{\langle 4, day\rangle, \langle 2, day \rightarrow stop\rangle, \left\langle 2, \overline{day} \rightarrow \overline{stop}\right\rangle\right\}$$

The penalties for each model are computed as follows:

- $pen_{GB}([day \wedge stop])$ = 0 + 0 + 0 = 0;
- $pen_{GB}([day \wedge \overline{stop}])$ = 0 + 2 + 0 = 2;
- $pen_{GB}([\overline{day} \wedge stop])$ = 4 + 0 + 2 = 6;
- $pen_{GB}([\overline{day} \wedge \overline{stop}])$ = 4 + 0 + 0 = 4;

Thus, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \succ [\overline{day} \wedge stop]$.

Frameworks based on the ***Hamming distance*** (Lafage and Lang, 2000) take into account the distance between goals and models, with the idea to favour models close to goals (and not to reason in binary fashion like in the previous kind of frameworks which only make a distinction between models satisfying a goal formula and models violating it). The distance from a model to a goal is the number of variables that must be flipped in the model in order to make it satisfy the formula. As in the previous frameworks, goals are associated with weights in a quantitative way. The preferred alternative is the one with the lower global distance calculated by summing over the goals the products of the elementary distance between the model and the goal and the weight associated with this goal.

Consider example 1.1 again to illustrate this formalism. The preferences of the agent can be expressed with the following set of goals GB where each goal is again a propositional formula with an associated weight.

$$GB = \left\{\langle 4, day\rangle, \langle 2, day \rightarrow stop\rangle, \left\langle 2, \overline{day} \rightarrow \overline{stop}\right\rangle\right\}$$

- For $M_1 = [day \wedge stop]$, we compute the distance from the model to each goal: d(M_1, day) = 0, d(M_1, $day \rightarrow stop$) = 0 and d(M_1, $\overline{day} \rightarrow \overline{stop}$) = 0 $\Rightarrow$ The resulting global distance from M_1 to GB is d(M_1, GB) = 4 x 0 + 2 x 0 + 2 x 0 = 0.

- For $M_2 = [day \wedge \overline{stop}]$:
  d(M_2, day) = 0, d(M_2, $day \rightarrow stop$) = 1 and d(M_2, $\overline{day} \rightarrow \overline{stop}$) = 0
  $\Rightarrow$ The resulting global distance from M_2 to GB is d(M_2, GB) = 4 x 0 + 2 x 1 + 2 x 0 = 2.

- For $M_3 = [\overline{day} \wedge stop]$:
  d(M_3, day) = 1, d(M_3, $day \rightarrow stop$) = 0 and d(M_3, $\overline{day} \rightarrow \overline{stop}$) = 1
  $\Rightarrow$ The resulting global distance from M_3 to GB is d(M_3, GB) = 4 x 1 + 2 x 0 + 2 x 1 = 6.

- For $M_4 = [\overline{day} \wedge \overline{stop}]$:
  d(M_4, day) = 1, d(M_4, $day \rightarrow stop$) = 0 and d(M_4, $\overline{day} \rightarrow \overline{stop}$) = 0
  $\Rightarrow$ The resulting global distance from M_4 to GB is d(M_4, GB) = 4 x 1 + 2 x 0 + 2 x 0 = 4.

Thus, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \succ [\overline{day} \wedge stop]$.

An other approach is the one of ***prioritized goals*** (Benferhat et al., 1993) where again preferences are expressed in term of propositional formulas corresponding to goals. This time, goals are associated with a function which gives the rank of each formula, a lower rank corresponds to a higher priority. Different algorithms are used to produce a preference relation from the information on goal ranking: "(i) the *bestout ordering*, focusing on the most prioritized violated goal, (ii) the *leximin ordering* which compares the cardinalities of satisfied goals at each level of priority, and (iii) the *discrimin ordering* which, when comparing two alternatives, does not take into account the goals satisfied by both." In this approach, numerical values are used in a more qualitative ways with a ranking objective. Especially Kaci and Prade (2008) show how to manage priorities with "symbolic" levels rather than numerical ones in a *possibilistic logic* manner (Dubois et al., 1994).

Consider example 1.1 again to illustrate this formalism. The preferences of the agent can be expressed with the following set of goals GB where each goal is a propositional formula with an associated rank.

$$GB = \langle \{G1, G2, G3\}, r \rangle$$

with G1 = day and r(1) = 1

G2 = $day \rightarrow stop$ and r(2) = 2

G3 = $\overline{day} \rightarrow \overline{stop}$ and r(3) = 2

According to the best-out ordering, we can compute the following ranks:

- For $M_1 = [day \wedge stop]$, $rank_{GB}(M_1) = min\,\{r(i)|M_1 \nvDash G_i\}$ = min $\emptyset$ = $+\infty$
- For $M_2 = [day \wedge \overline{stop}]$, $rank_{GB}(M_2) = min\,\{r(i)|M_2 \nvDash G_i\}$ = r(2) = 2
- For $M_3 = [\overline{day} \wedge stop]$, $rank_{GB}(M_3) = min\,\{r(i)|M_3 \nvDash G_i\}$ = r(1) = 1
- For $M_4 = [\overline{day} \wedge \overline{stop}]$, $rank_{GB}(M_4) = min\,\{r(i)|M_4 \nvDash G_i\}$ = r(1) = 1

Thus, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \sim [\overline{day} \wedge stop]$.

Frameworks based on ***conditional logics*** allow to express goals that the agent wants to satisfy only in a given context. Among them, we find preference relation based on *Z-ranking* (Pearl, 1990), *possibilistic logic* previously mentioned and *ceteris paribus* conditions (Wellman and Doyle, 1991) that we will later describe with CP-nets.

1.3.2.2 Constraint satisfaction

Constraint satisfaction (Bistarelli et al., 1999; Rossi et al., 2008) provides another approach that formalizes quantitative preferences but not qualitative ones. By considering "constraints" and "preferences" as related notions, different frameworks propose to model preferences as *soft constraints*. They distinguish requirements that cannot be violated (constraints) from requirements that can be violated even if this violation should be avoided as far as possible (preferences). When reasoning about the outcomes, constraints only have two possible status: being satisfied or violated while preferences are considered with more than just two levels of satisfiability (which may represent a violation cost or a level of importance or priority).

1.3.2.3 Graphical languages

Graphical languages are another approach to represent preferences. As for propositional logic languages, we can distinguish *qualitative models* that formalize ordinal preferences like LP-trees or CP-nets and *quantitative models* that formalize cardinal preferences like UCP-nets or GAI-networks.

Lexicographic preference trees (LP-trees) (Fraser, 1994; Flach and Matsubara, 2007) define an order of relative importance on the variables that describe the objects in a domain and the desirability of their values.

Let X_1,...,X_n be a set of boolean variables, such that the index represents the relative importance order. Let x_{i+} denote the preferred value of variable X_i. The simplest case of lexicographic ranking can be represented as an unlabelled binary decision tree with the following properties: (1) the only variable occurring at depth i is X_i – i.e., if an outcome is more important to a player, it appears higher in the preference tree; (2) in each split, the preferred value x_{i+} is the left branch. Consequently, the ranking order is represented by the left-to-right order of the leaves. Several classes of LP-trees exist where the importance relation of the variables and the desirability of their values can be conditional or unconditional (Booth et al., 2010).

Consider the simple example 1.1 again. The corresponding LP-tree is presented in Figure 1.6 and from the left-to-right order, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \succ [\overline{day} \wedge stop]$.

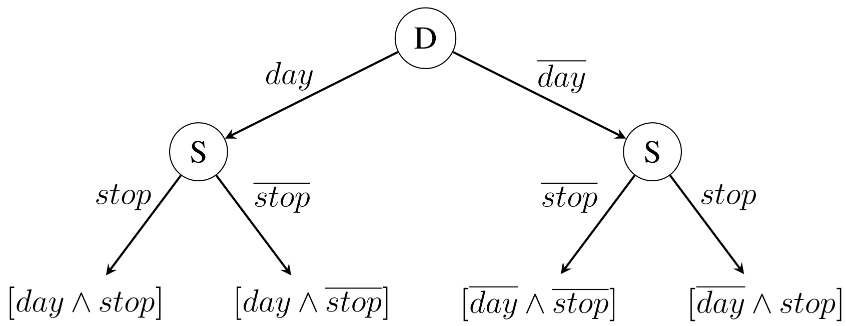


Figure 1.6: A lexicographic preference tree for the travel example.

CP-nets, ***conditional preference networks*** (Boutilier et al., 1999; Boutilier et al., 2004) are a graphical model that exploits *conditional preferential independence* to provide a compact representation of the preference order over all outcomes. Logically, CP-nets structure the decision maker's preferences under a *ceteris paribus* assumption: outcomes are compared, other things being equal. In words, CP-nets can formalize preferential dependencies with statements as "I prefer X_1 if Y_1, and X_2 if Y_2." where the values of the variable X depends on the values of the variable Y.

Consider the simple example 1.1 again. The corresponding CP-net is presented in Figure 1.7. An arrow between two nodes denotes a dependency between the variables and each node is associated with a table describing the user's preferences over the values of the variable given every combination of parent values.

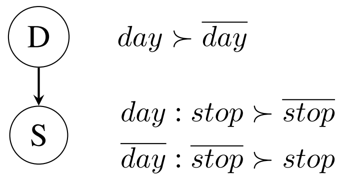


Figure 1.7: A CP-net illustration for the travel example.

Since in this model violating a preference of something on which your other preferences depend is worse than violating those other preferences, from CP-net 1.7, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \succ [\overline{day} \wedge stop]$.

TCP-nets, ***CP-nets with tradeoffs*** (Brafman and Domshlak, 2002) are an extension of CP-nets and also exploit *conditional preferential independence*. They complete the formalism by encoding information about *conditional relative importance* which allows to give different importance to different variables. Thus, TCP-nets combine aspects from the two previous formalisms : preferential dependencies used in CP-nets and relative importance used in lexicographic preferences trees. In words, TCP-nets can formalize statements as "It is more important to me that the value of X be high than that the value of Y be high".

Probabilistic CP-nets (Bigot et al., 2013) are a probabilistic extension of CP-nets and allow to formalize ill-known preferences (typically because they depend on the value of non controllable state variables, or because the system has few information about the user).

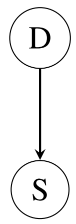
CI-nets, ***conditional importance networks*** (Bouveret et al., 2009) exploit the *ceteris paribus* interpretation and allow to express *importance statements* on arbitrary sets of variables, and not only on singletons as for the previous languages. CI-nets can formalize statements as "If I have a set A of goods, and I do not have any of the goods from some other set B, then I prefer the set of goods C over the set of goods D".

UCP-networks (Boutilier et al., 2001) are a directed graphical representation of conditional utility functions that combine aspects of CP-nets and of *Generalized Additive Independence* (GAI). UCP-nets extend CP-nets by allowing quantification of preferences with utilities. A utility function u has a GAI decomposition over $Z_1, ..., Z_k$ (not necessarily disjoint) subsets of V if $u(V) = \sum_{i=1}^{n} u_i(Z_i)$ for some functions u_i (Bacchus and Grove, 1995). The semantics in UCP-nets is given by

GAI along with the constraint that the directed influences reflect the **ceteris paribus** condition underlying CP-nets.

Consider the simple example 1.1 again. The preferences of the agent can be expressed with the UCP-net presented in Figure 1.8. The graph is a CP-net where each variable node is associated with a quantification (i.e., a set of factors $f_i(X_i, \mathbf{U}_i)$ where $\mathbf{U}_i$ are the parents of X_i).

day	$\overline{day}$
4	2



	$stop$	$\overline{stop}$
day	.9	.8
$\overline{day}$	.2	.3

Figure 1.8: A UCP-net illustration for the travel example.

According to UCP-net in Figure 1.8, we obtain the following utilities:

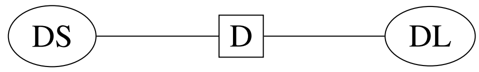
- $u([day \wedge stop]) = f_D(day) + f_S(stop, day) = 4 + 0.9 = 4.9$
- $u([day \wedge \overline{stop}]) = f_D(day) + f_S(\overline{stop}, day) = 4 + 0.8 = 4.8$
- $u([\overline{day} \wedge stop]) = f_D(\overline{day}) + f_S(stop, \overline{day}) = 2 + 0.2 = 2.2$
- $u([\overline{day} \wedge \overline{stop}]) = f_D(\overline{day}) + f_S(\overline{stop}, \overline{day}) = 2 + 0.3 = 2.3$

Thus, we obtain: $[day \wedge stop] \succ [day \wedge \overline{stop}] \succ [\overline{day} \wedge \overline{stop}] \succ [\overline{day} \wedge stop]$.

GAI-networks (Gonzales and Perny, 2004) are another graphical approach for the representation of quantitative preferences based on the *Generalized Additive Independence* (GAI) decomposition of utility functions and do not assume any CP-net structure.

Consider example 1.1 again. To illustrate GAI-nets, we complete the problem with a third variable L standing for the content of the luggage with two possible values $book$ and $music$. If the agent takes an overnight trip, he prefers to take music in his luggage, but during daytime he prefers a book.

The corresponding GAI-net is presented in Figure 1.9. The cliques (ellipse nodes) represent the group of dependant attributes and the separators (rectangle nodes) capture all the dependencies between sets of attributes.



$day\ stop$	10
$day\ \overline{stop}$	9
$\overline{day}\ stop$	1
$\overline{day}\ \overline{stop}$	2

$day\ book$	.9
$day\ music$	.8
$\overline{day}\ book$	.1
$\overline{day}\ music$	.2

Figure 1.9: A GAI-net illustration for the travel example.

According to GAI-net in Figure 1.9, we obtain the following utilities:

- $u([day \wedge stop \wedge book]) = u(day\ stop) + u(day\ book) = 10 + 0.9 = 10.9$
- $u([day \wedge stop \wedge music) = u(day\ stop) + u(day\ music) = 10 + 0.8 = 10.8$
- etc.

1.3.2.4 Our choice for preference modelling

Among these formalisms, we chose CP-nets (Boutilier et al., 2004) to model preferences in this work since they possess some clear advantages for modelling preferences derived from dialogues. In particular, they manage qualitative and conditional information and can deal with partial preferences which are unavoidable in spontaneous conversations. Chapter 6 below shows that CP-nets are a satisfactory option, since we manage to successfully formalize and reason about preferences represented by CP-nets. However, we do not pretend it is the only or even the best formalism to model preferences as they are naturally linguistically expressed and this work leaves open many other possibilities for studying other formalisms in connection with linguistically expressed preferences.

We give a detailed background about preference representation and reasoning with CP-nets in the following section.

1.3.3 CP-net, a model for qualitative preferences

As introduced in the previous section, CP-nets (Boutilier et al., 1999; Boutilier et al., 2004) are a graphical model that exploits *conditional preferential indepen-*

dence to provide a compact representation of the preference order over all outcomes. Logically, the CP-net structures the decision maker's preferences under a *ceteris paribus* assumption: outcomes are compared, other things being equal.

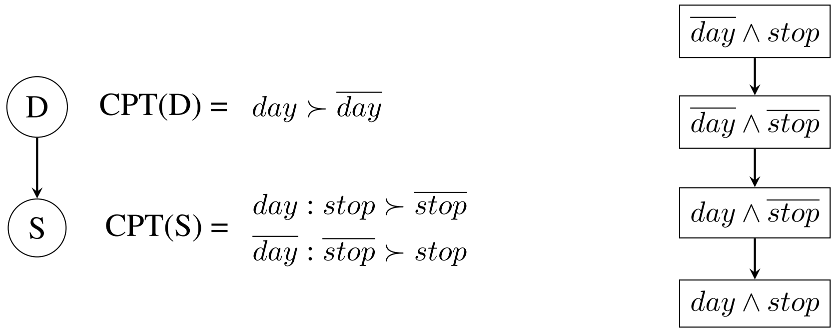
More formally, let V be a finite set of propositional variables whose combination of values determine all outcomes O. Then a *preference relation* $\succeq$ over O is a reflexive and transitive binary relation with strict preference $\succ$ defined in the usual way (i.e., $o \succeq o'$ and $o' \not\succeq o$). An agent is indifferent between two outcomes, written $o \sim o'$, if $o \succeq o'$ and $o' \succeq o$. Definition 1 defines *conditional preference independence* and Definition 2 defines CP-nets: the idea is that the graphical component $\mathcal{G}$ of a CP-net specifies for each variable $X \in V$ its *parent variables* $Pa(X)$ that affect the agent's preferences over the values of X, such that X is conditionally preferentially independent of $V \setminus (\{X\} \cup Pa(X))$ given $Pa(X)$.

Definition 1 *Let V be a set of propositional variables, each variable X_i with a domain $D(X_i)$. Let $\{X, Y, Z\}$ be a partition of V. X is* **conditionally preferentially independent** *of Y given Z if and only if $\forall z \in D(Z)$, $\forall x_1, x_2 \in D(X)$ and $\forall y_1, y_2 \in D(Y)$, $x_1y_1z \succeq x_2y_1z$ iff $x_1y_2z \succeq x_2y_2z$.*

Definition 2 *Let V be a set of propositional variables. $\mathcal{N}_{\mathcal{V}} = \langle \mathcal{G}, \mathcal{T} \rangle$ is a* **CP-net** *on V, where $\mathcal{G}$ is a directed graph over V, and $\mathcal{T}$ is a set of Conditional Preference Tables (CPTs). That is, $T = \{\text{CPT}(X_j)\colon X_j \in V\}$, where $\text{CPT}(X_j)$ specifies for each combination p of values of the parent variables $Pa(X_j)$ either $p : x_j \succ \overline{x_j}$, $p : \overline{x_j} \succ x_j$ or $p : x_j \sim \overline{x_j}$ where the $^-$ symbol sets the variable to false.*

Consider the simple example 1.1 again where the agent prefers to go from Paris to Hong Kong by day rather than overnight. If he takes an overnight trip, he prefers a nonstop flight, but if he goes by day he prefers a flight with a stop. The CP-net is presented in Figure 1.10(a).

The preference order over outcomes is recoverable from a CP-net. The logic for inferring this order abides by two ranked principles. The primary principle is that violating more preference statements is worse than violating fewer of them. The secondary principle is that violating a preference of something on which your other preferences depend is worse than violating those other preferences. Figure 1.10(b) shows the preference order over outcomes that follows from the CP-net in Figure 1.10(a) (each node is an outcome, corresponding to a complete assignment of the variables). An arc from outcome o_i to outcome o_j indicates a preference for o_j over o_i; so the top node is worst and the bottom one is best. For instance, $day \wedge \overline{stop}$ is preferred to $\overline{day} \wedge \overline{stop}$ because of the secondary principle in the logic: these outcomes each violate exactly one preference statement ($day : stop \succ \overline{stop} \in CPT(S)$ and $day \succ \overline{day} \in CPT(D)$ respectively), but by the secondary principle violating a statement in $CPT(D)$ is worse because D is a parent variable to S.



(a) (b)

Figure 1.10: CP-net and the induced preference graph for the travel example.

Because the CP-net in Figure 1.10(a) is acyclic, we can compute its optimal outcome using the linear *forward sweep* algorithm (Boutilier et al., 2004): this consists of instantiating variables following an order compatible with the graph, choosing for each variable (one of) its preferred values given the value of the parents. An iterated application of the algorithm, where one removes from the sample space the outcome that was identified as preferred in the last iteration, yields the relative preferences over all outcomes as shown in Figure 1.10(b).

1.4 Preferences in Natural Language Processing

Recall the systems for preference elicitation presented in Section 1.3.1. Even though these systems help users to find their hidden preferences or envisaging several attributes for which there is a need to consider trade-offs, in some cases they can be a constraint more than a help, especially when the interface is not intuitive and restrictive in the choice it allows to express. For example, consider a web interface for booking a flight. In general, the information about the preferences such a system allows an agent to express concern the dates and destination of the flight, sometimes the user may also specify if he wishes a non-stop flight and price category (economy, business or first class). But more specific constraints are almost always impossible to express, like wishing to find a trip for the cheapest price in a specific time range to any destination in a given country.

For those problems where the user wants to express specific constraints and in general for the majority of real life problems, we think it would be more intuitive and effective for the user to express his preferences in a natural language without any restriction of what a system allows to choose or not. This kind of approach would also resolve the problem these systems are confronted with, where the method needs to provide the right trade-off between *complexity* for which users prefer a simple system that minimise the time and efforts needed and *performance* for which more detailed and complete information will help to provide a more accurate response. Express their preferences in a natural language would prevent the user to provide effort in order to understand a specific interface and let them control the precision they are ready to provide. An other issue faced by existing preference elicitation systems is that their choice interface may influence users when constructing their preferences and we believe that a linguistic approach will lead to less influencing systems.

However, to our knowledge, there has been little work that has so far investigated how preferences can be efficiently elicited using Natural Language Processing (NLP) techniques. The *ExpertClerk* system (Shimazu, 2001) (see Section 1.3.1) is a first step towards a linguistic approach for preference elicitation but the natural language method it proposes is strongly dependent on the associated database. Some research in the field of NLP proposes another approach towards computational linguistic preference extraction with the study of comparative opinions (Liu, 2010). Comparative opinions express comparisons between two or more objects based on some of their shared features (e.g., "the picture quality of camera X is better than that of Y") (Jindal and Liu, 2006a; Jindal and Liu, 2006b; Ganapathibhotla and Liu, 2008). But this work is quite limited since it either only focuses on the task of identifying comparative sentences without extracting the comparative relations within the sentences, or when it does, it only considers comparisons at the sentence level, even sometimes with the assumption that there is only one comparative relation in a sentence. However, for reasoning with preferences, it is unavoidable to consider more complex comparisons with more than one dependency at a time and with a higher level than just the sentence in order to manage all the preference complexity.

1.5 Conclusion

In this chapter, we introduced some background about preferences by defining the most important notions we will use throughout the dissertation. We then gave an overview of previous work about preferences in the three different domain of research our work relates.

Game Theory is one of the first domains which became interested in preferences. Work in Game Theory proposes to capture preferences via expected utility, which is defined in terms of the agent's beliefs and utility function.

Since the 1990s, preferences became a research topic in *Artificial Intelligence* too. A lot has been done concerning preference acquisition, representation and reasoning. To acquire preferences, several systems have been proposed using preference learning where the preferences are implicitly learned from data describing the user's behaviour or past preferences and preference elicitation where preferences are the result of an interactive process with the user have been proposed.

However, to our knowledge, there has been little work that has so far investigated how preferences can be efficiently elicited from their linguistic expression. In *Natural Language Processing*, some work has recently been proposed to extract comparative opinions which is a first step towards preference extraction, however it only considers one comparison at a time and limits the study at the sentence level.

In this context, we propose a novel approach to extract preferences and model their evolution in spontaneous conversation using computational linguistic methods. As preferences are constructive and agents learn about the domain and even their own preferences during a decision process, we propose in the next chapter a study of preference change. We will use it in the following chapters to build an end-to-end method which extracts and reasons with verbalised preferences in dialogues.

Chapter 2

Preference change

2.1 Intrinsic and extrinsic preference changes 48

2.1.1 Extrinsic preferences aren't always calculable 49

2.1.2 Intrinsic preferences can change 54

2.1.3 Interim conclusion 58

2.2 CP-nets, a consistent model for preference change 59

2.3 Related work about preference change 65

2.3.1 Global decision models 66

2.3.2 Changing utilities or beliefs? 74

2.4 Conclusion 76

Rational choice in decision and game theory depends on preferences and beliefs (Simon, 1955). While philosophers and computer scientists have investigated belief change, preference change is less well studied. Game theory and decision theory regiment preferences via a static, pre-defined and complete utility function, which assigns each end state of the game or decision problem (and in some cases intermediate states too) a numerical reward. Most of these orthodox models assume that all possible states and actions are pre-defined and that preferences change only when beliefs do (see Section 1.2).

In Section 2.1, we introduce several natural decision and game problems that lack these features, arguing that they call for a *dynamic* rather than a static model of preferences: that is, preferences can change when unforeseen possibilities come to light or when there is no specifiable or measurable change in belief.

In Section 2.2, we develop an axiomatic and qualitative theory of preference change, in which we can model preference change due to various factors—not simply because beliefs change. Our model resembles AGM models of belief change (Alchourrón et al., 1985) and we argue that this is compatible with game theoretic approaches to rational action, increasing its coverage to a wider range of decision problems. We propose a formally precise dynamic model of preferences that extends and refines the existing static CP-net model of preferences (Boutilier et al., 2004). The axioms that update and revise preferences ensure that preferences remain consistent while minimising changes.

In Section 2.3, we explore the model's predictions by applying it to the examples from Section 2.1 that motivated the need for preference change. In particular, we study related work about preference change and show how our model bypasses Spohn's (2009) criticisms of the received models of preference change.

2.1 Intrinsic and extrinsic preference changes

Preferences can be *intrinsic* or *extrinsic* (Spohn, 2009). *Intrinsic* preferences are *sui generis*—they are not based on other considerations. They allow agents to prefer states that are inconsistent with reality or impossible to achieve; e.g., one can prefer to be skiing when one is actually in a meeting. The utility function in decision theory and game theory captures intrinsic preferences: it specifies the extent to which the agent finds a state attractive without regard to his beliefs about whether he can reach it or what subsequent states he could reach from it.

Extrinsic preferences depend on *external factors*, i.e., on other preferences together with *beliefs*: if X is preferred to Y because the agent believes that X makes achieving some other desirable outcome Z more likely, then X is *extrinsically* preferred to Y. For example, I could prefer eating fish over red meat not because I like fish better but because I believe my cholesterol is high and so I need to avoid eating meat. Decision theory and game theory capture extrinsic preferences via *expected utility*, which is defined in terms of the agent's *beliefs* and *utility function* (see section 1.2). We recall that the *expected utility* of an act $f \in F$ is $\sum_{s \in S} p(s).u(f(s))$ where p is the *probability function* over the set S of states of the world and u is the *utility function* over the set C of consequences. So actions that maximise expected utilities are an optimal trade off between what an agent would prefer to achieve and what he thinks he can achieve.

Expected utility captures the (uncontroversial) fact that extrinsic preferences are dynamic. As an agent's observations of the environment he's in change, his *probability function* change, so do his beliefs and hence also his expected utilities. However extrinsic preferences in classical game theory are dynamic *only* in virtue of the dynamics of the probabilistic belief model (Shoham and Leyton-Brown, 2009). The utility functions themselves do not change; they are static. Thus for classical theories, extrinsic preference change is entirely a matter of belief change and intrinsic preferences don't change. In this section we argue that this is not always the right picture—preferences may change in the absence of any measurable or specifiable belief change.

2.1.1 Extrinsic preferences aren't always calculable

We start with a practical motivation, exposing limitations in the current algorithms for computing expected utilities. Solving a decision or game problem amounts to finding actions that maximise expected utilities. In a game, we need to take account of other agents' reasoning, and so we have to optimize over strategies, which are functions from states to actions. Agents aim to identify *equilibrium strategies*, which are optimal in that no player would unilaterally deviate from one: each player expects at least as good a payoff from the actions that are specified in his own strategy when compared with any other strategy he could adopt, assuming that all the other players adhere to the strategies that are specified for them (Savage, 1954).[1]

Computing an equilibrium strategy depends on the preferences of every agent. But in many games an agent doesn't know the preferences of others. This uncertainty is modelled in game theory via a probability distribution over the possible *types* of players one is interacting with; each player type is associated with a complete and static utility function. So the dynamics of one's knowledge about another agent's preferences is still modeled purely via the dynamics of a probabilistic *belief* model rather than via a dynamic utility function.

All algorithms for identifying optimal actions, including approximations to game solutions such as *Monte Carlo Tree Search (MCTS)* which takes random samples in the decision space and builds a search tree according to the results (Browne et al., 2012), require right from the start of the game that all possible states and all possible actions are known, and that probability distributions over all possible outcomes of all possible actions are known too. In other words, any hidden information must be a foreseen possibility. In the tree built with the MCTS method, the values of intermediate states do not have to be evaluated but the value of the terminal state at the end of each simulation is required.

Let's see why this assumption is needed. Classical algorithms for identifying optimal actions use dynamic programming. For example, *backward induction*

[1] This model of decision making, where agents maximise expected utilities, is idealised: human behaviour often diverges from it (Kahneman and Tversky, 1979; Ariely, 2008; Yong and Xinlin, 2012). But we ignore this here since it has no bearing on the need to model preference change.

(Shoham and Leyton-Brown, 2009) starts by identifying for each penultimate state s an optimal action a for the last player p_l in the game. So a maximises p_l's expected utility, which is a function of his utility of the end states and the probability distribution of a's outcomes when it's performed in s, both of which are defined components of the game. Having identified p_l's optimal last action for each possible penultimate game state, one then calculates the expected utility of each penultimate state for *every* player, assuming that p_l plays an optimal action: again this calculation uses only defined components of the game. The algorithm then identifies the optimal penultimate move, using the penultimate player's expected utilities over penultimate states, which were just calculated in the previous step. This procedure continues until one reaches the initial state of the game, where one calculates the optimal first move on the basis of the first player's expected utilities of its outcomes (which were computed in the previous step, as before) (see Figure 2.1). Incomplete information and dynamic environments are handled by including probabilistic "moves by nature" within the game. For instance, in games where players lack complete information about the other agents' preferences, nature probabilistically assigns each player his type (Harsanyi, 1977). The dynamic programming algorithm takes these "nature moves" into account when calculating expected utility. Thus, *backward induction* requires knowing every possible sequence of moves that takes agents from the beginning to the end of the game to be enumerable *before* the first move in the game is chosen: this is because expected utilities are computed from the end of the game to the start of it, with every way of reaching one state from another being a part of the calculation.[2]

There are some games, however, where one cannot enumerate all ways of getting from the current state to an end state. Agents may be unable to fully identify the set of possible states in the game, or may lack information about what the possible actions are, or what actions the other players might contemplate performing in their own calculations of optimal play. In any of these cases, even if the intrinsic utilities over end states are known, agents cannot exploit the game tree because the *possibilities*—let alone their relative likelihoods—are unknown. This type of missing information is more serious than agents knowing the possibilities but being uncertain about reality, for which well known solutions exist. If there is missing information about the possibilities, *backward induction* on its own won't work. Neither will approximate methods, such as Monte Carlo Tree Search, for these assume that the parameters of the game are stationary during the sampling process, and they

[2]Some games, such as Texas Hold 'em Poker, are surveyable but too large to compute with effectively. A *preprocessing step* that converts the full game into a much smaller but strategically similar one makes computing optimal strategies feasible (Gilpin and Sandholm, 2007). But the algorithms for generating game abstractions also use dynamic programming working "backward" on the game tree, and so they require the full game description to be enumerable.

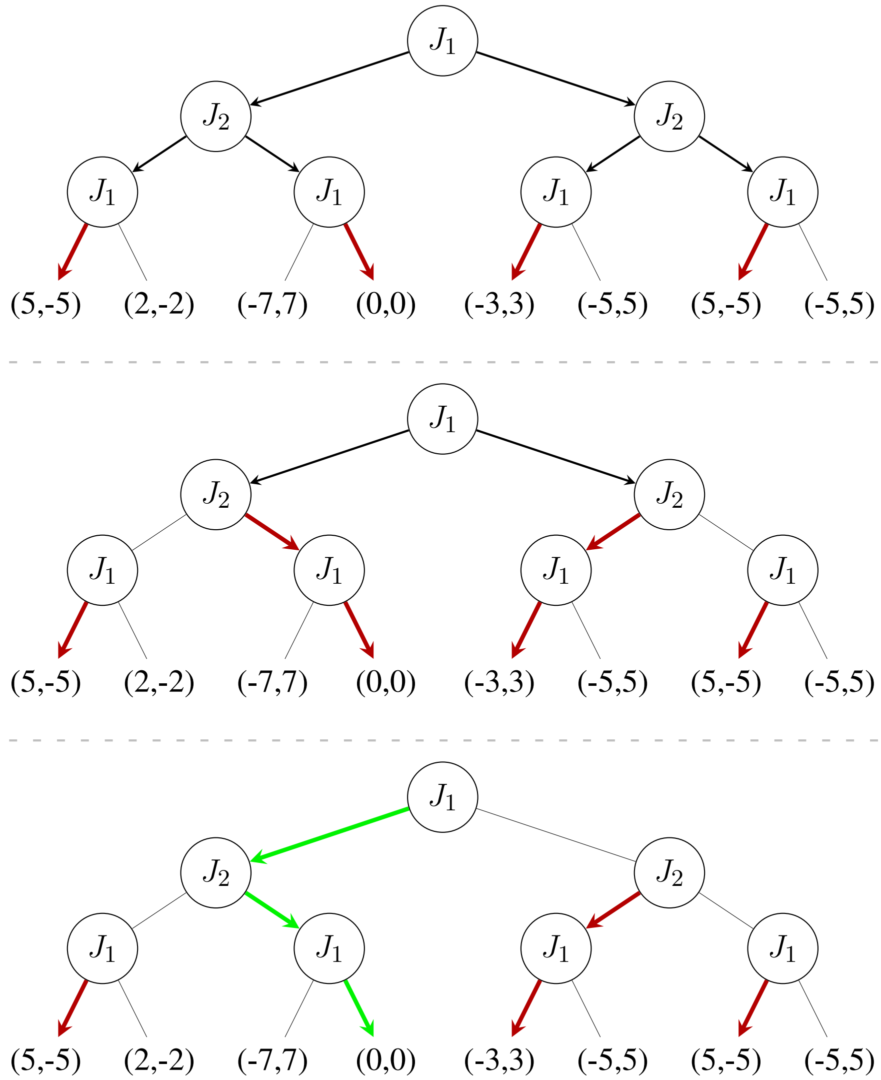


Figure 2.1: Backward induction algorithm.

Backward induction starts by identifying for each penultimate state s an optimal action a for the last player p_l in the game. So a maximises p_l's expected utility. Having identified p_l's optimal last action for each possible penultimate game state, one then calculates the expected utility of each penultimate state for *every* player, assuming that p_l plays an optimal action. This procedure continues until one reaches the initial state of the game, where one calculates the optimal first move on the basis of the first player's expected utilities of its outcomes (which were computed in the previous step, as before).

demand a complete joint probability distribution over all possible states and actions to be pre-defined.

One potential source of missing information is the possible actions that other players contemplate. This amounts to not knowing the set of possible player types and the probability distribution over them, which traditional models of games rely on. Arguably, the board game *The Settlers of Catan* (or *Settlers*) has this feature. Two to four players build settlements and cities connected by roads on the island of Catan. They must use certain resources (clay, ore, sheep, wheat and wood) in order to build; e.g., a road requires 1 clay and 1 wood. Victory points get awarded to players in several ways: e.g., by building a settlement (1 point) or a city (2 points). It is a win lose game: the first player with 10 victory points wins.[3] There are several thousand end states but it is always clear who wins; so there is common knowledge about each player's intrinsic preferences. But players often negotiate trades with one another in order to obtain the resources they need. Players can agree to any trade, which makes the game tree non-enumerable: there are an unbounded number of possible trades because agents can promise to perform a particular future move as a part of the trade, e.g., *If you trade clay for wood now, I will give you wheat when I get it* (see Section 3.1.2 for the description of a corpus of humans playing *Settlers*). Natural language also provides an unbounded way of expressing such trades.[4] They can lie or bluff, too.

Since negotiations make the game tree non-enumerable, players must optimize on a sub-part of it. But there is no common knowledge as to which subpart of it each player has isolated so as to perform his calculations, let alone what preferences each player adopts for the states in that subpart. Thus every player has insufficient knowledge of his opponents to model them as a probability distribution over a set of possible player types—they do not know which actions are a part of the opponents' decision procedures.

All is not lost, however. A player can use standard algorithms to optimize over a *subpart* G' of the whole *Settlers* game, so long as he has relatively accurate ways of estimating the parameters of G' that are required by these algorithms. In particular, he needs a method for identifying which intermediate states of the whole *Settlers* game are "end" states to G', and ways of estimating preferences over those states.

How might a player calculate which end states in G' put him in a strong position to win the game? Unable to use backward induction or Monte Carlo Tree Search, he could estimate preferences on the basis of games similar to the one he is playing and for which he has the information (perhaps with hindsight) relevant to

[3] See `http://www.catan.com/` for a more detailed description of the rules

[4] *Settlers* has been tackled quite successfully using Monte Carlo Tree Search (Szita et al., 2010). But this work is applied to simplified versions of the game where agents don't negotiate trades.

computing estimated utilities—in particular, complete information about the possible player types for those similar games. Alternatively, he might exploit similarity metrics between intermediate states and winning end states.However the agent estimates the preferences that define the subgame G', it is *these* rather than the intrinsic preferences over Settler's end states that are treated as *direct input* to calculations of optimal action via *backward induction* or its approximations. In effect, the agent treats the intermediate states as end states.

Crucially, all methods for estimating the preferences of intermediate states involve ways of defeasibly inferring beliefs about how intermediate states relate to end states. In the previous paragraph, we mentioned two ways to calculate defeasible links; no doubt there are others. However one does the calculation, it is defeasible because the inference necessarily makes default assumptions about the missing parts of the game tree. When an agent learns more about the game (in this example, he learns more about his opponents' possible player types and the actions that they contemplate), the defeasible inferences change. Thus the preferences over the "end" states of G' must be subject to change and revision as the agent learns more about the parameters of G' through his play and observations.[5] So the agent must calculate optimal action from a *dynamic* function from states to preferences as well as from a dynamic model of belief; he cannot rely on a static and complete model of intrinsic preferences. Our agent isn't exploiting the complete game tree to find optimal actions, but (at most) only the part of it that he can foresee (Asher et al., 2013) (see Figure 2.2).

Because players in *Settlers* lack information about the extrinsic preferences that their opponents are (currently) trying to optimize over, they often engage in what game theorists call *pre-bargaining chat*—they elicit the preferences of others, for example, asking *Do you need wood?* or *Will you exchange wood for clay?* Agents also reveal their own preferences; e.g., by saying *I need clay*. Because the game is large and because elicitation has costs of various sorts (e.g., costs stemming from deception and politeness (Asher and Quinley, 2011)), the elicitation process will yield only a partial and fallible picture of the agents' preferences. So players must use preference information that's inferred through conversation to update and revise their model of their opponents: this is often the best evidence they have about what actions their opponents are considering, and with what utilities, as shown in the dialogue (2.1), taken from a corpus of humans playing *Settlers* (Afantenos et al., 2012b) (see Section 3.1.2):

[5]This is distinct from updating reward functions during reinforcement learning (Sutton and Barto, 1998). Reinforcement learning acquires the parameters of a game by playing it many times; in our example, the agent has only one opportunity to play the game, and he must learn it while playing it just that one time.

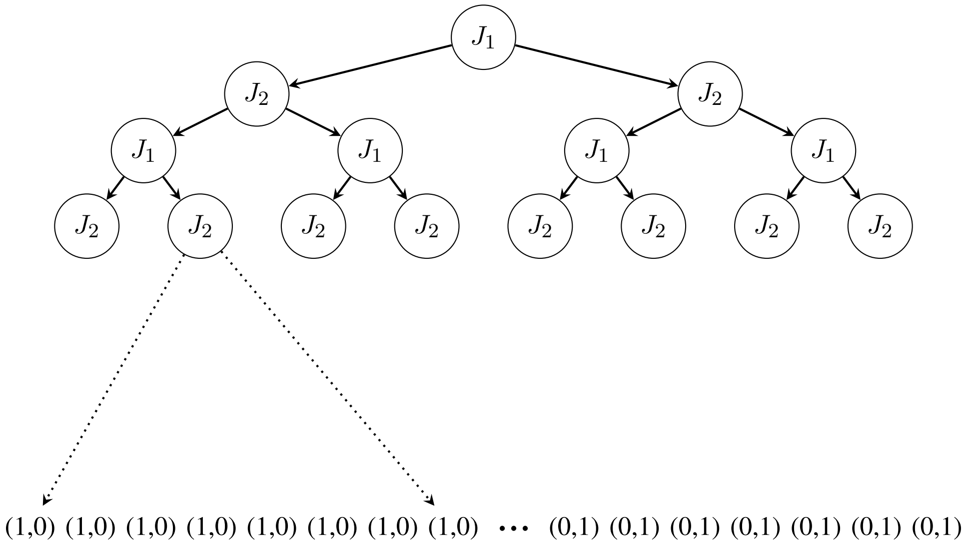


Figure 2.2: Partial reasoning in big game.
At the beginning of the game, there are so many possible states that agents can't reason over the whole game. So instead they reason to find optimal actions over subgames.

(2.1)
π_1 G: to return, I was proposing a wheat for a sheep, if you will take 1

π_2 D: ah yes, i had been trying to say that i would've needed 2 wheat, but now i've changed my plan anyway

π_3 G: so you now need no wheat, or you have changed your plan to take 1?

π_4 D: I won't be needing it.

D declares in π_2 that his preference for wheat has changed from what it was, prompting a clarification question from G. These dynamic preferences cannot effectively be calculated from a pre-existing model of intrinsic preferences and beliefs because, as we said, there's no way of reliably enumerating which belief hypotheses are relevant.

2.1.2 Intrinsic preferences can change

In Section 2.1.1, we discussed *The Settlers of Catan*, a game where agents are forced to act without a complete definition of the game. This compels players, we argued, to use fallible calculations of extrinsic preferences over intermediate states, which will often change in the light of new evidence. In this section, we provide several examples of *intrinsic* preference changes, especially change due to the discovery of new state descriptions and change in the absence of any belief change or new discovery.

2.1.2.1 Preference change due to the discovery of new state descriptions

Suppose an agent has an intrinsic preference to buy a new car, but he doesn't know what car models are available. So at the start of the game the agent has a preference ordering over *state descriptions* that don't discriminate among all the individual end states themselves—he lacks sufficient domain knowledge to know what those end states are. As the agent learns about what end states are possible, he will refine his preferences over them, creating a more fine-grained partition. Indeed, the relative preferences over two specific states may change as his preferences over state descriptions change: while at the start of the process the agent might prefer any state where he has a car to one where he doesn't, he may subsequently discover one car model that in his view is so bad that he would prefer to have no car at all over a car of that model!

Given his lack of domain knowledge, the agent can plan to talk to a car salesman. More generally, one can plan for conditional actions, and make decisions about subsequent actions contingent on their outcome. Such a staged approach to identifying the optimal strategy is made explicit in Hierarchical Task Network (HTN) Planning (Erol et al., 1994). But conditional actions within these frameworks deal with foreseen but hidden contingencies, equipping the agent with the option of finding out information he deems relevant to his decision problem. Planning, game theory and decision theory aren't equipped to handle the "unknown unknowns", cases where an agent does not know what he doesn't know. None of these frameworks handle the discovery of an entirely new *concept*, or more technically the discovery of a *novel state description* or *novel action description*. They assume a fixed language. In the case we consider here, the language describing the problem is expanded, because there are unforeseen ways of defining the decision problem.

Discovering new state descriptions can occur in mundane and natural ways. We take an example from cooking. Suppose that an agent has never heard of the spice turmeric, nor has he tasted it or cooked with it. And suppose that initially he isn't all that keen on pumpkin soup; he would rather eat fish. Someone then persuades him to try a pumpkin soup, which he finds delicious, far better than any pumpkin soup that he has tasted before. He asks about its ingredients and discovers a new type of food, turmeric, to which he takes an immediate liking; so much so that he now prefers to cook with it whenever possible. Importantly, his *intrinsic* preferences change. Specifically, now that he's discovered turmeric, and found that it goes so well with pumpkin, he would rather have pumpkin soup than any alternative food, even fish which he preferred previously. More technically, there are possible end states that are a part of his decision problem of which he was initially totally unaware (i.e., states where he has pumpkin soup with turmeric), and these states turn out to be most preferred. Crucially, the new utility function isn't necessarily an *extension* of the old one—in our example, all states where the agent eats fish initially had a higher utility than those where he eats pumpkin soup, but eating fish now has a lower utility than eating pumpkin soup *unless* turmeric isn't available.

An agent who wishes to buy a car may at the start lack information entirely about the existence of certain car features—for instance, Anti-lock Brake Systems (ABS). He learns about ABS from the car salesman, and on discovering its existence he decides that this feature is highly desirable. Indeed, thanks to his discovery of an entirely new concept—ABS—his *intrinsic* preferences change. More technically, there are possible end states that are a part of his decision problem of which he was totally unaware (i.e., states where he owns a car with ABS), and these states turn out to be most preferred. Crucially, the new utility function isn't necessarily an *extension* of the old one—discoveries can trigger *revisions* to existing intrinsic preferences. For instance, before discovering ABS, he may have had an intrinsic preference for a cheap car with high emissions over an expensive one with low emissions. In other words, price was more important to him than environmental factors, although ideally he would like a cheap car with low emissions. But after discussions with the salesman about ABS, he may now have decided that safety is paramount, whatever the price. So his intrinsic preferences for a cheap car have been revised: all cheap cars are no longer preferred to all expensive ones. This in turn affects his extrinsic preferences: while at the start of his actions he preferred car a to car b on the grounds that it is cheaper and with lower emissions, on learning that b has ABS, he now prefers it to car a.

This example shows that new options that an agent had not foreseen can present themselves while he is acting in the environment. Desires are parasitic on beliefs in that one can't prefer an object to alternatives or desire an object if one has no idea of that object; there is no *de re* desire without *de re* beliefs (Asher, 1987). Agents must therefore formulate their relative preferences over new concepts as and when they're discovered and not before, and in doing so they may *revise* (and not merely extend) existing preferences over the already known possibilities.

Arguably, this is preference revision in the face of changed beliefs—it was triggered by the discovery of unforeseen possibilities.But existing game or decision theoretic models, such as *Markov Decision Processes (MDPs)* (Bellman, 1957), don't handle this sort of belief revision. MDPs deal with sequential decision problems in fully observable environments. They model belief as a probability distribution over a pre-defined set of possible states, and neither this set nor dependencies between them change over time. But in our example, the discovery of *novel state descriptions* leads the agent not only to update his belief model to include new random variables and/or new values for existing variables, but potentially to revise their dependencies as well. MDPs likewise characterize intrinsic preferences with a static utility function. But in our example, discovering novel variables for describing the game prompts the agent to update his intrinsic preferences, with preferences over existing factors potentially being *revised* rather than extended (e.g., the relative preference for fish over pumpkins has changed). MDPs don't capture this sort of preference dynamics either. In essence, these Markovian models of decision making don't cover situations where the description of the game is changing while the agent is playing it.

2.1.2.2 Preference change in the absence of any belief change

We've so far discussed examples where an agents intrinsic and extrinsic preferences change with changes in belief, in particular with the discovery of unforeseen possibilities. We now argue that agents can change their intrinsic preferences in the absence of any new discovery or belief change.

Suppose that an agent A would intrinsically prefer to smoke than not. More formally, let the propositional variable s stand for "A smokes" and $\overline{s}$ its negation, and let $\succ$ be the preference relation $\succ$. So $s \succ \overline{s}$. Moreover, A may intrinsically prefer a peaceful, quiet and healthy life (p) over a non-peaceful life in which his family and friends nag him ($\overline{p}$). So $p \succ \overline{p}$. These preferences generate a moral dilemma, because he also believes that his smoking (defeasibly) implies a non-healthy and non-peaceful life—he is fully aware of the health risks and his family will nag him about his smoking! That is, where the formula $\phi > \psi$ means "If ϕ then normally ψ", the agent believes $s > \overline{p}$. So his two global intrinsic preferences for s and p cannot be reconciled, given his beliefs.

Suppose A decides, perhaps on the basis of rational deliberation, to resolve this moral dilemma. It is definitely *not* rational deliberation that compels him to resolve the dilemma one way as opposed to another. In particular, it is not rational deliberation that compels him to give up smoking! To put this another way, by deciding to give up smoking, A has essentially decided that his preference for p is more important to him than his preference for s. But there is no way of deciding on the relative weighting of p vs. s in the resolution of the moral dilemma: you cannot explain why p won purely on the basis of the beliefs and the equally weighted preferences in the agent's first state.

More formally, A's global intrinsic preference for p over $\overline{p}$ is unchanged, but he abandons the *global* intrinsic preference $s \succ \overline{s}$ for smoking. It is now dependent on p: given p, he prefers $\overline{s}$; if p is false, however, then he would still prefers s because he still craves a cigarette! One can express such dependencies among preferences with a formalism as the one of CP-nets (see section 1.3.3): $p : \overline{s} \succ s$ means "if p, $\overline{s}$ is preferred over s". So the preferences have changed from a pair of global preferences to those in (2.2):

$$\begin{aligned} & p \succ \overline{p} \\ p: \ & \overline{s} \succ s \\ \overline{p}: \ & s \succ \overline{s} \end{aligned} \tag{2.2}$$

A then changes even further. After months of not smoking, his desire to smoke has changed into an aversion to smoking. At this stage, *even if* his belief $s > \overline{p}$ were to change—perhaps he loses touch with his family and friends and there's new evidence that smoking is healthy—he still prefers not smoking. In other words, his preference for $\overline{s}$ that was conditional on his preference for p has now become a *global* preference for $\overline{s}$. So the intrinsic preferences have changed again, from (2.2) to (2.3): he doesn't want to smoke any more, in any circumstances.

$$\begin{aligned} p \succ \overline{p} \\ \overline{s} \succ s \end{aligned} \tag{2.3}$$

This is an example of intrinsic preference change without rational deliberation, knowledge discovery or belief change. It happens simply through a change in taste, perhaps borne from a change in habits. Other examples of this kind include: vegetarians who initially give up meat for health or environmental reasons, who miss bacon sandwiches at first but who over time lose their taste for meat; and children who change their tastes as they mature.

2.1.3 Interim conclusion

We have presented a number of decisions and game problems where agents' preferences change. We've argued that preference change in these scenarios cannot be defined entirely in terms of belief change, and so go beyond traditional models of rational action.

Preference change is an irreducible component of these decision and game problems, and so reasoning about action must be able to handle dynamic preferences as well as dynamic beliefs. Making preferences dynamic in their own right paves the way for supporting inferences about optimal action in games where existing algorithms for computing extrinsic preferences break down. It is also needed for expressing the effects of discovering unforeseen possible actions and states during game play. And finally, preferences can change in the absence of any belief change or any new discovery.

2.2 CP-nets, a consistent model for preference change

In this section and the following, we want to show how CP-nets, a compact representation of preferences and their dependencies (see section 1.3.3), are suitable for modelling preference change. Especially, we want to show how consistency is preserved when we perform a preference revision, we want to guarantee that the preference relation remains consistent if the original CP-net was consistent.

If the preference model is updated with a new preference that is consistent with it, then simply adding the new preference is unproblematic. The challenge is in defining update with a new preference that is inconsistent with the existing preferences. To ensure that the updated model is consistent, we need some notion of *preference revision*, analogous to the notion of *belief revision*. Following the AGM model (Alchourrón et al., 1985) (see appendix A.1 for a description of this framework), we define preference revision as a sequence of two operations: 1) *downdating* the existing preferences to a *maximal subset* of them that are consistent with the new preference, followed by 2) *adding* the new preference to the result (so new preferences take priority over old ones).

One of the contentious areas of belief revision is how to downdate a belief model when the maximal subset of old beliefs that are consistent with the new one is not unique. In this case theories exploit some notion of *entrenchment* (Gärdenfors and Makinson, 1988): a transitive, binary relation on propositions where "p is more entrenched than q" means that p is of more "epistemic value", making an agent more reluctant to give up his belief in it (all else being equal). Intuitively, the more entrenched propositions are more useful in deliberation; e.g., p is a natural law whereas q is a contingent fact. So when more than one maximal subset of the old beliefs is consistent with the new beliefs, one favours retaining a maximal subset where the least entrenched propositions are removed.

Since we're making preference revision analogous to AGM belief revision, we also need a concept that's analogous to entrenchment, to help an agent decide which maximal subset of old preferences to retain. Fortunately, unlike the modal logic of belief, CP-nets have an explicit partial order—the graphical model—that helps solve this problem. Recall that CP-nets define which variables influence the preferences over other variables. In effect, the structure of the CP-net defines which preferences are global and which are derivative on other preferences. So we can regiment the intuition that the fewer factors there are that compel us to have a particular preference, the more entrenched that preference is and the less prepared we are to give it up (unless failure to do so results in abandoning more preferences overall). This priority for removing derivative preferences over global ones bears similarities to the secondary principle in the logic of CP-nets for inferring the preference order over all outcomes: i.e., it is worse to violate a preference on variables over which your other preferences depend.

Accordingly, we form a partial order over the outcomes that are defined by a CP-net, which reflects the extent to which an agent would be prepared to give up his preference for one outcome as compared with giving up his preference for another outcome. If the agent's preferences for outcomes o_i and o_j are dependent on each other, then the agent is equally reluctant to give up either of them. On the other hand, if the preference for o_i depends on a superset of the factors on which the preference for o_j depends, then the agent is more reluctant to give up his preferences for o_j. We call this partial order the *preference surrender value* or PSV (we use the term surrender, an antonym of entrenchment, because we'll assign numeric PSVs to outcomes where the higher the number, the more inclined one is to give up a preference for it). Definition 3 defines the partial order PSV in two steps. First, it detects cyclically dependent outcomes in the CP-net $\mathcal{N}_V$ and constraints their PSVs to be equal. This forms a partition over outcomes. Then it assigns each element in each partition its numeric PSV value: the lower the number the *less* one is inclined to give up the preference:

Definition 3 *The partial order of* ***preferential surrender value*** *(or PSV) over the variables V in a CP-net $\mathcal{N}_V$ is defined as follows:*

1. *For $V' \subseteq V$ such that $\mathcal{N}_V|V'$ describes cyclically dependent preferences over V', we say that $PSV_{\mathcal{N}_V}(o_i) = PSV_{\mathcal{N}_V}(o_j)$ for all $o_i, o_j \in V'$ (so o_i and o_j are state descriptions that assign specific values to each of the variables $a_1, \ldots, a_n$ in V').*

2. *With V thus partitioned into equivalence classes of cyclically dependent outcomes, we assign each equivalence class a numeric preference surrender value or* PSV *as follows:*

   - *For $V_0 \subseteq V$ such that V_0 is an equivalence class of outcomes such that all preference statements in $\mathcal{N}_V$ about outcomes in V_0 depend only on elements in V_0 or none at all, we set $PSV_{\mathcal{N}_V}(o_i) = 0$, for all $o_i \in V_0$*

- *For any V_n, $n \neq 0$, such that V_n is an equivalence class such that all preference statements in $\mathcal{N}_V$ about outcomes in V_n depend on $\vec{x}$, where $\vec{x} \notin V_n$, we set PSV$_{\mathcal{N}_V}(o_i) = 1 + max\{PSV(x)\colon parent(x, o_i) \wedge x \notin V_n\}$, for all $o_i \in V_n$*

So suppose a (perhaps partial) CP-net $\mathcal{N}$ is updated with a new preference statement $\phi : R(t, t')$, where $R \in \{\prec, \succ, \sim\}$. To maintain consistency, one first checks whether $\mathcal{N}$'s transitive closure entails $\phi : \overline{R(t, t')}$, where $\phi : \overline{R(t, t')}$ is equivalent to $t \prec t' \vee t \sim t'$ if $R(t, t')$ is $t \succ t'$. If so, then we must change or *reset* formulae in $\mathcal{N}$ so that the result together with $\phi : R(t, t')$ is consistent. Following earlier discussion, the ranking in Definition 4 favours those resets with the fewest changes to preferences of any outcomes, and more changes to outcomes with a larger PSV than a smaller PSV.

Definition 4 *The* $>$***-ranking over Resets*** *is defined as follows:*
$Reset_n(\mathcal{N}) > Reset_m(\mathcal{N})$ *iff*

- $Reset_n(\mathcal{N})$ *resets fewer equations in* $\mathcal{N}$ *than* $Reset_m(\mathcal{N})$*,*
- *or, they reset the same number of equations, and:*
  min{PSV(o): R(o,o') is reset by Reset$_n$(N) } > min {PSV(o): R(o,o') is reset by Reset$_m$(N)}

We can now stipulate that during preference revision, any downdating is restricted to Resets of $\mathcal{N}$ that are consistent with the new preference and $>$-maximal. We obtain the following definition where $*$ is the *revision* operator ($\mathcal{N} * \phi$ means that ϕ is added to $\mathcal{N}$ and at the same time other statements are removed if this is needed to ensure that the resulting CP-net is consistent) and $+$ is the *expansion* operator ($\mathcal{N} + \phi$ means that ϕ is added to $\mathcal{N}$ without checking the consistency: nothing is removed).

Definition 5 *The* **preference revision** $\mathcal{N} * \phi$ *is defined as follows:*

$$\mathcal{N} * \phi = \begin{cases} \bigcap\{Reset_i(\mathcal{N})\colon Reset_i(\mathcal{N})\ is\ > -maximal\ and \\ \qquad Reset_i(\mathcal{N}) + \phi\ consistent\} + \phi, \text{if } \phi \text{ is consistent} \\ \perp \text{ otherwise} \end{cases}$$

Let X be a variable with domain $D(X) = \{a, b, c\}$. We illustrate Definition 5 with the following example, in which the partial CP-net (2.4) is updated with $c \succ a$.

$$CPT(X) = \begin{array}{l} a \succ b \\ b \succ c \end{array} \tag{2.4}$$

All preferences in the CP-net (2.4) have rank 0 because our one variable has rank 0. So there are two minimal resets of (2.4) that are consistent with $c \succ a$, given in (2.5) and (2.6):

$$CPT(X) = \begin{array}{l} a \succ b \\ \mathbf{c \succ b} \end{array} \quad (2.5) \qquad CPT(X) = \begin{array}{l} \mathbf{b \succ a} \\ b \succ c \end{array} \quad (2.6)$$

In words, (2.5) retains $a \succ b$ and resets $b \succ c$ to $c \succ b$, whereas (2.6) retains $b \succ c$ and resets $a \succ b$ to $b \succ a$. Since their intersection is empty, updating (2.4) with $c \succ a$ yields only the new information $c \succ a$.

On the other hand, consider CP-net (2.7) with two variables X_1 and X_2 whose domains are $D(X_1) = \{a, \overline{a}\}$ and $D(X_2) = \{b, c, d\}$. Updating this CP-net with $\overline{a}\colon d \prec c$ leaves $a \prec \overline{a}$ and the preferences dependent on a intact.

$$\begin{aligned} CPT(X_1) =\ & a \prec \overline{a} \\ CPT(X_2) =\ & a : b \prec c \\ & a : c \prec d \\ & \overline{a} : c \prec b \\ & \overline{a} : b \prec d \end{aligned} \tag{2.7}$$

Lemma 1, which follows from Definition 5, shows that preference revision is relatively well-behaved.

Lemma 1

1. *Success:* $\phi \in \mathcal{N} * \phi$ *(and so trivially* $\mathcal{N} * \phi \models \phi$*).*
2. *Inclusion:* $\mathcal{N} * \phi \subseteq \mathcal{N} + \phi$ *(that is, the deductive or transitive closure of* $\mathcal{N} * \phi$ *is contained in that of* $\mathcal{N} + \phi$*).*
3. *Vacuity: If* $\neg\phi \notin \mathcal{N}$ *and* ϕ *is consistent, then* $\mathcal{N} * \phi = \mathcal{N} + \phi$.
4. *Consistency:* $\mathcal{N} * \phi$ *is consistent if* ϕ *is consistent.*
5. *Extensionality: If* $\vdash \phi \leftrightarrow \psi$*, then* $\mathcal{N} * \phi = \mathcal{N} * \psi$.

Compared to the six basic Gärdenfors postulates for belief revision in the AGM model (see Appendix A.1), we don't want the *Closure* postulate because we reason with partial descriptions of preferences. We think it is a fact of life that preference information is usually incomplete and we are interested in the process of reasoning with incompleteness. From this major point, our model for preference change based on CP-nets differs from the one of Hansson (1995) which only considers preferences that refer to complete alternatives (i.e., to elements of a set of mutually exclusive alternatives). CP-nets also present the advantage of providing a notion of importance among the preferences thanks to the dependency structure. This allows to reason with the notion of entrenchment which is not handled by Hansson's model.

Definition 5 handles all the types of intrinsic preference changes that we detailed in Section 2.1. Assuming that one can infer a preference from the agent's current action, Definition 5 provides the means to update one's model of that agent's preferences with that newly inferred preference. So we can compute (consistent) partial preferences from observing an agent's behaviour. In addition, since our model of partial preferences has a model theory that's defined in terms of complete preferences, where the latter support a logic for identifying the optimal action, we can also predict the agent's optimal actions from our partial model: one simply completes the partial model of preferences by defaulting to indifference for the preference information that is missing entirely, and one uses the resulting complete representation to infer what decision the agent will make next.

We now analyze the scenarios of intrinsic preference change from Section 2.1 when the agent discovers new possible outcomes. In all cases, the agent starts with a partial CP-net $\mathcal{N}_V$ that induces a partial order over all state descriptions that use V. In the turmeric example, the agent learns of a new possibility, expressed via a new state description. In effect he changes from playing a game using V to playing a game using V' where $V \subseteq V'$. Thus the partition of states into distinct state descriptions based on V is refined into a partition based on V'. For example, consider an example inspired by Section 2.1 about discovering a new type of food. Suppose that the agent initially prefers to eat fish over pumpkin soup, and he prefers coffee to tea. His initial vocabulary V consists of two variables D for the drink with domain $D(D) = \{\mathit{coffee}, \mathit{tea}\}$ and M for the meal with domain $D(M) = \{\mathit{fish}, \mathit{soup}\}$. And his (partial) CP-net $\mathcal{N}_V$ is (2.8):

$$\begin{aligned} CPT(D) = & \quad \mathit{coffee} \succ \mathit{tea} \\ CPT(M) = & \quad \mathit{fish} \succ \mathit{soup} \end{aligned} \tag{2.8}$$

The agent then discovers a new type of food, changing his vocabulary to V'. We will illustrate different effects on preference revision with different examples for V'.

Suppose first that the preferences over $V' - V$ are *independent* from those over V: for all $t \in V' - V$, $a, b \in V$ if $t \wedge a \succ t \wedge b$ then $\bar{t} \wedge a \succ \bar{t} \wedge b$. Then, it is immediate from Definition 5 that all prior preferences in $\mathcal{N}_V$ persist in $\mathcal{N}_{V'}$. So suppose the agent discovers chocolate, extending V to a new vocabulary V' with a variable

C with domain $D(C) = \{chocolate, \overline{chocolate}\}$, and he loves it, preferring to eat it whatever the context—i.e., he acquires the new preference *chocolate* $\succ$ $\overline{\textit{chocolate}}$. By Definition 5 the preferences in (2.8) persist in his updated preferences (2.9):

$$\begin{aligned} CPT(D) = &\quad \textit{coffee} \succ \textit{tea} \\ CPT(M) = &\quad \textit{fish} \succ \textit{soup} \\ CPT(C) = &\quad \textit{chocolate} \succ \overline{\textit{chocolate}} \end{aligned} \tag{2.9}$$

On the other hand, suppose that there are preferences in V that are *dependent* on the new preferences in $V' - V$. For example, suppose the agent discovers wine and turmeric such that V' contains two new variables W for the preference over wine with domain $D(W) = \{\textit{white\_wine}, \textit{red\_wine}\}$ and T for the preference over turmeric with domain $D(T) = \{\textit{turmeric}, \overline{turmeric}\}$. Suppose that he prefers *white_wine* with fish, but *red_wine* with soup, and he now prefers *soup* over *fish* whenever *turmeric* is available, and he prefers *turmeric* over $\overline{\textit{turmeric}}$, whatever the circumstances. Then according to Definition 5, updating the CP-net (2.8) with this new vocabulary and preference information yields the preference statements in (2.10):

$$\begin{aligned} CPT(D) = &\quad \textit{coffee} \succ \textit{tea} \\ CPT(T) = &\quad \textit{turmeric} \succ \overline{\textit{turmeric}} \\ CPT(M) = &\quad \textit{turmeric} : \quad \textit{soup} \succ \textit{fish} \\ &\quad \overline{\textit{turmeric}} : \quad \textit{fish} \succ \textit{soup} \\ CPT(W) = &\quad \textit{fish} : \ \textit{white\_wine} \succ \textit{red\_wine} \\ &\quad \textit{soup} : \quad \textit{red\_wine} \succ \textit{white\_wine} \end{aligned} \tag{2.10}$$

Specifically, Definition 5 entails that the same preferences are retained for old variables that are independent of the new preference statements (like *coffee* and *tea*). For those new variables that are dependent only on variables in $\mathcal{N}_V$ (*white_wine* and *red_wine*), Definition 5 simply adds these new preference statements to the updated CP-net; they do not trigger revision to the existing preference statements. But if the *old* variables are dependent on the *new* ones (like *fish* and *soup*, whose preferences now depend on the new variable *turmeric*), revision is more complex because the new preference statement may be inconsistent with the existing ones. Computing the result involves exploring the recursive structure of the CP-net. The simplest case is where $\mathcal{N}_V$ is without dependencies, binary comparisons only, no indifference and:

- $t \in V' - V$, $a, b \in V$ with $a \succ b \in \mathcal{N}_V$ but $t\colon b \succ a \in \mathcal{N}_{V'}$.

The question is: should the preference on a and b in $\mathcal{N}_{V'}$ also shift to $b \succ a$, given $\bar{t}$? Of course they can, but this is equivalent to a shift in preferences among a and

b within $\mathcal{N}_V$ itself, because $(\vec{x} \wedge t\colon \phi) \wedge (\vec{x} \wedge \bar{t}\colon \phi)$ is semantically equivalent to $\vec{x}\colon \phi$. In other words, the agent's preference change between a and b would in this case be *independent* of the new vocabulary (or discovery) t. Instead, it's an intrinsic preference change that is specifiable within the "smaller" vocabulary (and hence holds within a domain of fewer possibilities), as defined by V. But the new preference information in our example does not invoke such an intrinsic preference change on the old vocabulary; it only specifies how the preferences among fish and soup change when the (newly discovered) turmeric is available. Accordingly, Definition 5 yields the CP-net (2.10), where the old preference for *fish* over *soup* is retained in the context $\overline{\textit{turmeric}}$.

More generally, consider an arbitrary binary comparison $\mathcal{N}_V$ with no indifferences, and imagine the introduction of preferences over outcomes describable in $V' - V$. The new preferences over $V' - V$ will have a rank in $\mathcal{N}_{V'}$:

Lemma 2 *Equations in $\mathcal{N}_V$ persist within $\mathcal{N}_{V'}$ for all pairs of outcomes o_1, o_2 where:*

1. *$PSV(o_1), PSV(o_2) > PSV(t)$, for all $t \in V' - V$*
2. *if $\vec{x}\colon R(o_1, o_2)$ and $\vec{x} \wedge t\colon \overline{R(o_1, o_2)}$, then $\vec{x} \wedge \bar{t}\colon R(o_1, o_2)$*

Introducing indifference into the scenario complicates matters, because given $t\colon o_1 \succ o_2$ in $\mathcal{N}_{V'}$ and $o_1 \prec o_2$ in $\mathcal{N}_V$, then it is still possible to have $\bar{t}\colon o_1 \sim o_2$ in $\mathcal{N}_{V'}$. But that means that we just slightly weaken Lemma 2 to account for indifference:

- If $\vec{x}\colon R(o_1, o_2)$ and $\vec{x} \wedge t\colon \overline{R(o_1, o_2)}$, then $\vec{x} \wedge \bar{t}\colon R(o_1, o_2) \vee o_1 \sim o_2$

Now consider the smoking example. The agent starts with a CP-net consisting of $s \succ \bar{s}$ and $p \succ \bar{p}$. To resolve his moral dilemma, the agent changes his tastes: now p is more preferred than s rather than equally preferred; i.e., he updates his preferences with $p : \bar{s} \succ s$. According to Definition 5, the result retains the preference for s in the context $\bar{p}$, as shown in (2.2). The agent then changes taste again, preferring $\bar{s}$ over s whatever the circumstances. Updating (2.2) with $s \succ \bar{s}$ yields (2.3) according to Definition 5.

2.3 Related work about preference change

Previous work on preference change has focused on classifying preference change into different types based on etiology: preference change due to belief change, to a change in taste (Bradley, 2007), to changes in the environment (Lang and van der Torre, 2008), or to a change in the order of importance of constraints over fixed outcomes (Liu, 2008). This work has largely concentrated on extrinsic preference change. Our work complements this earlier research; we have motivated and modeled both intrinsic and extrinsic preference change.

Grune and Hansson (2009b) distinguish several models of preference change, namely: derivational, temporal, consistency-preserving and evolutionary models. *Derivational* models focus on the interaction between preference change and belief change. *Temporal* models are about how preference change over time. *Consistency* models investigate consistency preservation, and *environmental* models focus on the evolution of preferences in multi-agent environments. Lemma 1 shows our formalisation of preference change preserves consistency. In this section, we show how our approach can also formalize derivational and temporal problems. First, we present a formalisation of Spohn's extrinsic and intrinsic temporal examples (Spohn, 2009). We then consider Hill's "sour grapes" problem (Hill, 2009) as an example of a derivational model (see Section 2.3.2).

2.3.1 Global decision models

The only approach of which we're aware for modelling intrinsic preference change within standard decision or game theory is to construct a decision problem or game over a set of games—what Spohn (2009) calls a *global decision model*. According to Spohn, all received models for handling (foreseen) preference change can be articulated as a global decision model. Informally, a global decision model is a standard decision tree with one crucial difference: in addition to chance and end nodes and actions that form links between nodes, it contains *agent nodes*, with each agent node being the root of a different local decision model. Each agent node has its own set of actions, probability distributions over their outcomes, and utilities over the end nodes. All of these components can vary from one agent node to another. The outcome of an action can be a chance node, agent node or end node. Thus a global model can represent foreseen preference change, with the agent choosing actions that affect the type of agent he will be (note that it cannot deal with unforeseen preference change of the kind we discussed in Section 2.1). Intrinsic preference change corresponds to the local decision trees that are rooted at two distinct agent nodes having distinct utilities over the same end state. Extrinsic preference change corresponds to the local decision trees having distinct actions and/or distinct probabilities over their outcomes.

Spohn shows that global decision problems don't provide sufficient information for identifying optimal strategies. He does this via two minimal pairs of decision problems: one pair involves extrinsic preference change; the other pair involves intrinsic preference change. For each minimal pair, he shows that both problems in the pair have isomorphic global decision models, but intuitively their optimal strategies are distinct. We show here that our framework doesn't suffer from the same shortcomings: our models of Spohn's decision problems have distinct optimal strategies.

2.3.1.1 Two examples of extrinsic preference change

In both examples, the agent must choose an action that has a good or a bad outcome depending on chance, and before choosing this action he can choose an option that may change his beliefs about the action's consequences. The first story, which we call **EPC-a** (EPC for Extrinsic Preference Change), starts with an agent deciding whether to refuse (g_1) or take (g_2) a test for diagnosing if he has a serious disease (b_1) or not (b_2). The agent then faces a further decision: to take treatment (h_1) or not (h_2). The treatment cures the disease with no side effects if the agent has the disease, but it has unpleasant side effects if he doesn't have the disease. If untreated, the disease has even more adverse effects than the side effects.

Now, actions g_1 vs. g_2 yield distinct sorts of agents, each with distinct probabilities on whether they have the disease. There are *four* different types of agents: the first agent a_1 results from the action g_2—refusing the test. a_1 is uncertain about whether he has the disease, with an even probability distribution over b_1 vs. b_2—i.e., $P_1(\vec{B}) = \langle 0.5, 0.5 \rangle$. Action g_1 (taking the test) has an indeterminate outcome. There is a 50% chance of reaching certainty about the disease, with equal chances for positive and negative diagnoses—agent types a_2 and a_3 respectively, so $P_2(\vec{B}) = \langle 1, 0 \rangle$ and $P_3(\vec{B}) = \langle 0, 1 \rangle$. There is also a 50% chance that that the test is mute, leaving the resulting agent a_4 just as uncertain about $\vec{B}$ as a_1 (i.e., $P_4(\vec{B}) = \langle 0.5, 0.5 \rangle$). All four agents a_1 to a_4 have the same options available to them—to take treatment (h_1) or to refuse it (h_2)—and the same utilities over the end states. Spohn assigns the following plausible utilities across all four agent types: $U(h_1, b_1) = U(h_2, b_2) = 2$ (in both these cases the agent is healthy and without side effects); $U(h_1, b_2) = -2$ (the agent suffers side effects of treatment); and $U(h_2, b_1) = -10$ (the untreated disease is much worse than the side effects).

The extrinsic preferences (or expected utilities) among the four agent types differ because their probability distributions over $\vec{B}$ differ. Intuitively, it should be clear that the optimal strategy is to take the test (g_2): this gives the agent a 50% chance of being in a state where taking any risks over the decision to have treatment is unnecessary, and if taking that 50% chance doesn't pay off then the risk is no worse than that of refusing the test. The representation of this decision problem is shown in Figure 2.3. The local decision problems T_1 to T_4 for the agents a_1 to a_4

all have the structure and utilities shown in the subtree on the right, but vary in the probability distribution of the outcomes of the actions h_1 and h_2.

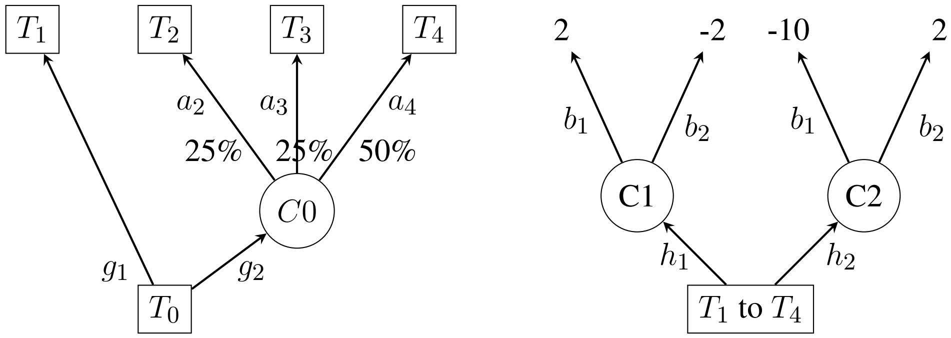


Figure 2.3: The global decision models for EPC-a and EPC-b.

Now consider Spohn's alternative scenario **EPC-b**. An agent must catch a train, which might leave at 8am (b_1) or at 11am (b_2). If the agent goes early to the station (h_1) he runs the risk of waiting for 3 hours, but if he goes late (h_2) he might miss the train. The utilities over $\vec{H} \times \vec{B}$ from EPC-a are thus highly plausible for EPC-b too. Prior to deciding what time to turn up, the agent can either choose to write down the possible departure times (g_1) or not (g_2). If he writes down the times (becoming agent a_1), then he remains equally uncertain about the train's arrival time. On the other hand, if he does not write down the two times, there is a 50% chance that he will remain uncertain about the arrival time (agent a_4) and also a 50% chance that he will forget one of the times, with an evens chance that the time he forgets is 8am (agent a_3) or 11am (agent a_2). So the probability distributions over $\vec{B}$ for agents a_1 to a_4 are also identical to those given in EPC-a. Thus the global decision model for EPC-b is also that given in Figure 2.3. But intuitively, the optimal initial decision, between g_1 and g_2, is different from EPC-a: the agent should write down the two possible departure times (g_1), so as to guarantee that his uncertainty about the departure times is preserved, thereby compelling him to leave early (h_1) instead of running the risk of missing the train. Thus in EPC-a and EPC-b, the optimal strategies are intuitively different, but the isomorphic global decision models can't distinguish them.

Our approach differs from Spohn's and the classical frameworks that he criticises. Specifically, CP-nets have an additional expressive power, allowing agents to have preferences over *intermediate states* as well as end states. Thus we can distinguish EPC-a from EPC-b in spite of their identical preferences over end states and probability transitions: our representation of EPC-a will include preference statements in the CP-net that stipulate that taking the test is preferred over not taking it because of the benefit this affords, at an intermediate node, of the possibility of acquiring knowledge. Global decision models can't express such statements. The benefits among the intermediate nodes of EPC-b are different and so our representation of this scenario won't be isomporphic to that of EPC-a.

The CP-net for representing EPC-a is as follows. The **intrinsic preferences** are defined in terms of variables C and S. The domain $D(C)$ of C is $\{\mathit{cured}, \overline{\mathit{cured}}\}$, where *cured* stands for the cases where the agent is cured, either because he was not suffering from the disease or because he was but he took the treatment and so is cured. $D(S) = \{\mathit{sideEffects}, \overline{\mathit{sideEffects}}\}$, where *sideEffects* occurs as a result of treatment in the absence of disease and $\overline{\mathit{SideEffects}}$ occurs either because the agent didn't take the treatment or he did but he also had the disease. **Extrinsic preferences** exist over the means for reaching the satisfaction of intrinsic preferences (the agent has control over these means in this scenario). There are two variables, Te and Tr, where $D(Te) = \{\mathit{test}, \overline{\mathit{test}}\}$, and $D(Tr) = \{\mathit{treat}, \overline{\mathit{treat}}\}$. Preferences on these variables depend on the **state of the world** (that the agent does not fully control). In particular, it depends on whether the agent thinks he's ill, healthy or ignorant about it, and (so) it depends also on the results of the test, where t_p, t_n and $t_?$ stand respectively for a positive, a negative and an indeterminate diagnosis.

We will approximate the probabilistic belief model of EPC-a with a qualitative model. The formula $\mathcal{B}(\mathit{ill})$ means that the agent believes he's ill and $\overline{\mathcal{B}(\overline{\mathit{ill}})}$ means that the agent does not believe that he is not ill. The **axioms** in (2.11) capture Spohn's domain-level descriptions concerning the results of the test and of treatment:

$$
\begin{aligned}
& test \rightarrow (t_p \vee t_n \vee t_?) \\
& test \wedge t_p \leftrightarrow \mathcal{B}(\mathit{ill}) \\
& test \wedge t_n \leftrightarrow \mathcal{B}(\overline{\mathit{ill}}) \\
& (test \wedge t_?) \vee \overline{\mathit{test}} \leftrightarrow \overline{\mathcal{B}(\mathit{ill})} \wedge \overline{\mathcal{B}(\overline{\mathit{ill}})}
\end{aligned}
\tag{2.11}
$$

In words, the agent believes he's ill if and only if he takes a test that yields a positive result, he believes he isn't if and only if he takes a test that yields a negative result, and otherwise he is ignorant about it. We assume that all these axioms are believed where belief is S4; so, for example, it follows that $\mathcal{B}(t_p)$ entails $\mathcal{B}(\mathit{ill})$. We also assume that test and its results are all observable—e.g., $\mathit{test} \leftrightarrow \mathcal{B}(\mathit{test})$ and $t_p \leftrightarrow \mathcal{B}(t_p)$. And finally, the test results t_p and t_n are completely reliable (and the agent knows this): in other words, $\mathcal{B}(\mathit{ill}) \rightarrow \mathit{ill}$ and $\mathcal{B}(\overline{\mathit{ill}}) \rightarrow \overline{\mathit{ill}}$.

Spohn's example is complex, with extensive interactions between beliefs and preferences. We formalize the preferences in this story with the following (partial) CP-net. First, that our agent prefers to be cured over not being cured, and that he prefers no side effects to side effects are both independent, global preferences (indeed, they are intrinsic preferences though our model only reflects this indirectly).

However, he is willing to take the treatment (and so risk side effects) in cases where he takes no test or the test provided an indeterminate diagnosis—i.e., when he is ignorant about his health, $\overline{\mathcal{B}(\textit{ill})} \wedge \overline{\mathcal{B}(\overline{\textit{ill}})}$, which we'll abbreviate as *Ign*.

$$CPT(C) = \textit{cured} \succ \overline{\textit{cured}} \qquad \textbf{(intrinsic preference)}$$

$$CPT(S) = \overline{\textit{sideEffects}} \succ \textit{sideEffects} \qquad \textbf{(intrinsic preference)}$$

$$CPT(Te) = \textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \overline{\mathcal{B}(\textit{ill})} \wedge \overline{\mathcal{B}(\overline{\textit{ill}})} : \textit{test} \succ \overline{\textit{test}}$$

$$\begin{aligned} CPT(Tr) = {} & \textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \textit{test} \wedge t_p : \textit{treat} \succ \overline{\textit{treat}} && \text{(i)} \\ & \textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \textit{test} \wedge t_n : \overline{\textit{treat}} \succ \textit{treat} && \text{(ii)} \\ & \textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \textit{test} \wedge t_? : \textit{treat} \succ \overline{\textit{treat}} && \text{(iii)} \\ & \textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \overline{\textit{test}} : \textit{treat} \succ \overline{\textit{treat}} && \text{(iv)} \end{aligned}$$

The intuitively optimal strategy is derivable from the above CP-net as follows. Assuming that the agent starts in a state of *Ign*orance about whether he has the illness or not, the forward sweep algorithm makes $\textit{cured} \wedge \overline{\textit{sideEffects}} \wedge \textit{test}$ a part of any solution to the above preferences. But then the axioms (2.11) ensure that the value for *treat* is dependent on the results of *test*. So in words, the agent's optimal strategy is to take the test and to make a subsequent decision on *treat* dependent on its outcome.

Our model ensures that the ordering over the end states matches Spohn's ordering. That is, $((\textit{ill} \wedge \textit{treat}) \sim (\overline{\textit{ill}} \wedge \overline{\textit{treat}})) \succ (\overline{\textit{ill}} \wedge \textit{treat}) \succ (\textit{ill} \wedge \overline{\textit{treat}})$ follows given our exogenous representation of the domain. Our proof that this follows rests on the semantics of preference statements: $p \succ q$ means that each state that satisfies p is preferred to at least some state that satisfies q. Given domain-level axioms in (2.11) and the logic of belief and of preferences:

from (i), we obtain: $(\mathcal{B}(\textit{ill}) \wedge \textit{treat}) \succ (\mathcal{B}(\textit{ill}) \wedge \overline{\textit{treat}})$ (v)
from (ii), we obtain: $(\mathcal{B}(\overline{\textit{ill}}) \wedge \overline{\textit{treat}}) \succ (\mathcal{B}(\overline{\textit{ill}}) \wedge \textit{treat})$ (vi)
from (iii) and (iv), we obtain: $(\textit{Ign} \wedge \textit{treat}) \succ (\textit{Ign} \wedge \overline{\textit{treat}})$ (vii)

And since the test results are observable and reliable:

from (v), we obtain: $(\textit{ill} \wedge \textit{treat}) \succ (\textit{ill} \wedge \overline{treat})$ (viii)
from (vi), we obtain: $(\overline{\textit{ill}} \wedge \overline{\textit{treat}}) \succ (\overline{\textit{ill}} \wedge \textit{treat})$ (ix)

We will now show how to derive $\overline{\textit{ill}} \wedge \textit{treat} \succ \textit{ill} \wedge \overline{\textit{treat}}$. The crucial clause is (vii)—in ignorance the agent still prefers treatment. By the axioms on preferences:

from (vii) we obtain: $((\mathit{ill} \vee \overline{\mathit{ill}}) \wedge Ign \wedge \mathit{treat}) \succ ((\mathit{ill} \vee \overline{\mathit{ill}}) \wedge Ign \wedge \overline{treat})$ (x)
from (x), we obtain:
$\mathit{Ign} : ((\mathit{ill} \wedge \mathit{treat}) \vee (\overline{\mathit{ill}} \wedge \mathit{treat})) \succ ((\mathit{ill} \wedge \overline{\mathit{treat}}) \vee (\overline{\mathit{ill}} \wedge \overline{\mathit{treat}}))$ (xi)

Furthemore, we assume that ignorance does not make one alternative that is dispreferred in the case of complete information more preferred: that is (xii) and (xiii) hold too, thanks to (viii) and (ix) respectively:

$\mathit{Ign} : (\mathit{ill} \wedge \mathit{treat}) \succ (\mathit{ill} \wedge \overline{\mathit{treat}})$ (xii)
$\mathit{Ign} : (\overline{\mathit{ill}} \wedge \overline{\mathit{treat}}) \succ (\overline{\mathit{ill}} \wedge \mathit{treat})$ (xiii)

from (xi) and preference semantic, we obtain:
$\mathit{Ign} : (\overline{\mathit{ill}} \wedge \mathit{treat}) \succ ((\mathit{ill} \wedge \overline{\mathit{treat}}) \vee (\overline{\mathit{ill}} \wedge \overline{\mathit{treat}}))$ (xiv)
and from (xiii) and (xiv) and the semantics, we obtain:
$\mathit{Ign} : (\overline{\mathit{ill}} \wedge \mathit{treat}) \succ (\mathit{ill} \wedge \overline{\mathit{treat}})$ (xv)

Our partial preference model doesn't validate any relative ordering between the outcomes $\mathit{ill} \wedge \mathit{treat}$ and $\overline{\mathit{ill}} \wedge \overline{\mathit{treat}}$. So as we mentioned earlier, completing the preference model so as to decide on optimal action involves defaulting to indifference among these two outcomes (for in the absence of any information, the preference relation defaults to indifference). So this indifference plus (xii), (xiii) and (xv) yields Spohn's ordering over his end states, as given by his utility function in Figure 2.3.

We'll now represent the second story **EPC-b** as a (different) partial CP-net and belief statements. The variables $D(T) = \{\mathit{train}, \overline{\mathit{train}}\}$ (catch the train vs. miss it) and $D(W) = \{\mathit{wait}, \overline{\mathit{wait}}\}$ (wait, or not) define the **intrinsic preferences**; the variables $D(Wr) = \{\mathit{write}, \overline{\mathit{write}}\}$ (write the train times, or not) and $D(G) = \{\mathit{early}, \mathit{late}\}$ (go to the station early, or late) the **extrinsic preferences**. These preferences depend on the **states of the world** described in (2.12), namely: if the agent *writes* then he retains uncertainty about the training leaving at *8am* or *11am*, while $\overline{\mathit{write}}$ leads to three different states f_8, f_{11} and f_n which respectively mean the agent forgets the $8am$ departure time, the $11am$ departure time, or neither:

$$\begin{aligned}
&\overline{\mathit{write}} \rightarrow (f_8 \vee f_{11} \vee f_n)\\
&\overline{\mathit{write}} \wedge f_8 \leftrightarrow \mathcal{B}(11am)\\
&\overline{\mathit{write}} \wedge f_{11} \leftrightarrow \mathcal{B}(8am)\\
&(\overline{\mathit{write}} \wedge f_n) \vee \mathit{write} \leftrightarrow \overline{\mathcal{B}(8am)} \wedge \overline{\mathcal{B}(11am)}
\end{aligned} \tag{2.12}$$

We can formalize this story with the following CP-net:

$$CPT(T) = train \succ \overline{train} \qquad \textbf{(intrinsic preference)}$$

$$CPT(W) = \overline{wait} \succ wait \qquad \textbf{(intrinsic preference)}$$

$$CPT(Wr) = train \wedge \overline{wait} \wedge \overline{\mathcal{B}(8am)} \wedge \overline{\mathcal{B}(11am)} : write \succ \overline{write}$$

$$\begin{aligned} CPT(G) = \; & train \wedge \overline{wait} \wedge write : early \succ late && \text{(i)} \\ & train \wedge \overline{wait} \wedge \overline{write} \wedge f_8 : late \succ early && \text{(ii)} \\ & train \wedge \overline{wait} \wedge \overline{write} \wedge f_{11} : early \succ late && \text{(iii)} \\ & train \wedge \overline{wait} \wedge \overline{write} \wedge f_n : early \succ late && \text{(iv)} \end{aligned}$$

This CP-net validates the optimal action *write* and the same preferences over outcomes as Spohn's: $(((8am \wedge early) \sim (11am \wedge late)) \succ (11am \wedge early) \succ (8am \wedge late))$. Specifically, given the semantics of beliefs and preferences and the axioms (2.12):

from (iii), we obtain: $(8am \wedge early) \succ (8am \wedge late)$ (v)
from (ii), we obtain: $(11am \wedge late) \succ (11am \wedge early)$ (vi)
from (i) and (iv), we obtain:
$((8am \vee 11am) \wedge early) \succ ((8am \vee 11am) \wedge late)$ (vii)
from (vii), we obtain:
$(11am \wedge early) \succ ((8am \wedge late) \vee (11am \wedge late))$ (viii)
from (vi) and (viii), we obtain: $(11am \wedge early) \succ (8am \wedge late)$

Thus, completing the CP-net by defaulting to indifference on missing preference information yields: $((8am \wedge early) \sim (11am \wedge late)) \succ (11am \wedge early) \succ (8am \wedge late)$.

2.3.1.2 Two examples of intrinsic preference change

In both scenarios in this minimal pair, the agent forms a preference at an "initial" time, and he can either decide to act on it (pre-empting future "agents" with perhaps distinct preferences from performing actions) or he can wait and see whether time changes his preference. The first story **IPC-a** is one where he starts with a preference to go on holiday (b_1) over not going (b_2). But first, he must choose between going immediately to the travel agent to book (a_1), or to "sleep on it" until the next morning (a_2), to see whether he still thinks that b_1 is worth the money it's going to cost him. There is no objective way of deciding which of a_1 vs. a_2 is optimal. But intuitively, it seems reasonable for the agent to mistrust his excitement now (especially given the cost) and to sleep on it; i.e., a_2 is intuitively optimal.

The second story, **IPC-b** is one where the agent is at a market and he is offered goods that look cheap but ornate. He believes that they are never worth the money that the hawker is selling them for, nor even the money that the hawker might sell them for at the end of a long bargaining process. So initially, the agent prefers not to buy (b_1) over buying (b_2). However, the hawkers are very persistent. The agent must either close his mind (a_1) to their offers and thus stick to his initial preference for b_1, or he must listen to them (a_2), and risk that they talk him into preferring b_2 over b_1. This time, intuitively the optimal action is to ignore the hawker; i.e., a_1 is optimal over a_2. As before, IPC-a and IPC-b have isomorphic global decision models (see Figure 2.4), which provide insufficient information for distinguishing their diverging optimal strategies.

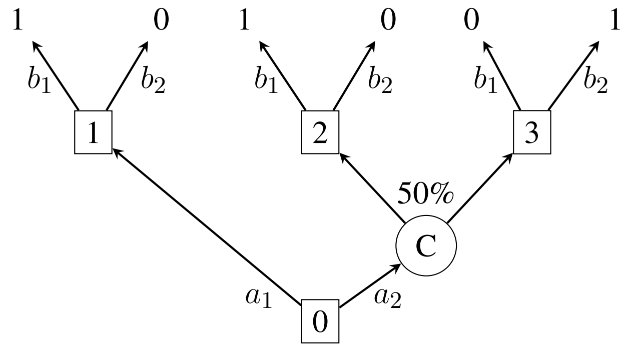


Figure 2.4: The global decision model for IPC-a and IPC-b.

Let's now represent these stories in our framework. Implicit in Spohn's formulation of the decision problems is the idea that the agent should not be impulsive: his decision to buy should be the result of deliberation that is as objective as he can make it. We make this explicit in our representation via a variable R, where $D(R) = \{\textit{reason}, \overline{\textit{reason}}\}$. The agent also has (intrinsic) preferences over the variable B, where $D(B) = \{\textit{buy}, \overline{\textit{buy}}\}$.

These two stories also involve **extrinsic preferences**. For each story there is only one variable. For IPC-a the variable W takes the value that the agent *wait* before deciding whether to buy the product, or not wait ($\overline{\textit{wait}}$). For IPC-b the variable L takes the value that the agent *listen* to the hawker before deciding whether to buy, or not listen ($\overline{\textit{listen}}$). Story IPC-a is then captured with the following *initial* preference statements:

$$CPT(R) = reason \succ \overline{reason} \qquad \textbf{(intrinsic preference)}$$

$$CPT(B) = buy \succ \overline{buy} \qquad \textbf{(intrinsic preference)}$$

$$CPT(W) = reason \wedge buy : wait \succ \overline{wait}$$

The preference over W captures the intuition that when an agent wants to be reasonable and also wants to (currently) buy the product, he prefers to wait to see if he changes his mind. After waiting, either the preferences over B stay the same, or there is a preference change, thereby effecting a change to $CPT(B)$: the CP-net below results from updating via Definition 5 the original CP-net with the new (global) preference $\overline{buy} \succ buy$.

$$CPT(R) = reason \succ \overline{reason} \qquad \textbf{(intrinsic preference)}$$

$$CPT(B) = \overline{buy} \succ buy \qquad \textbf{(intrinsic preference)}$$

$$CPT(W) = reason \wedge buy : wait \succ \overline{wait}$$

In contrast, story IPC-b is captured with the following preference statements:

$$CPT(R) = reason \succ \overline{reason} \qquad \textbf{(intrinsic preference)}$$

$$CPT(B) = \overline{buy} \succ buy \qquad \textbf{(intrinsic preference)}$$

$$CPT(L) = reason \wedge \overline{buy} : \overline{listen} \succ listen$$

This preference over L means that the agent chooses $\overline{listen}$, and so there is no opportunity for the hawker to persuade the agent to change his preferences over B.

Our analyses of Spohn's examples show that we can better predict optimal action in the face of foreseen preference change. This is because we don't model all possible choices for an agent in all the possible present and future states. Instead, we model the agents' preferences only at any given moment and allow that model to be updated over time according to the evidence that's observed. For instance, intrinsic preference change triggers a downdate to the CP-net (see Definition 5). In contrast, Spohn's model of intrinsic preference change involves no downdating or revision of an initial preference model. On the other hand, our analysis of EPC-a involves no downdating, but rather predicts a (potential) change in whether *treat* is a part of the optimal action sequence, stemming from a belief change that's caused by the (initial) optimal action to take the *test*.

2.3.2 Changing utilities or beliefs?

Hill (2009) distinguishes intrinsic from extrinsic preference change by studying the "sour grapes" problem based on a La Fontaine fable: a fox attempts to get some grapes in a tree; realizing that he cannot reach them he turns away, saying to himself that they are sour. Hill illustrates changes in utilities and beliefs by considering a second situation where the fox spots a ladder and takes the decision to try again or to give up.

The story involves an **intrinsic preference** over a variable G, standing for having the *grapes*, or not ($\overline{\textit{grapes}}$). The set of **acts** A takes the values to attempt to *get* the grapes or to *walk* away. Preferences over these acts correspond to Spohn's **extrinsic preferences**. Finally, the story implies some **beliefs** about the world. The fox's preferences depend on the following properties of any given state: (a) whether the grapes are *low* or high (i.e., $\overline{\textit{low}}$); (b) whether there is a *ladder* or not (i.e., $\overline{\textit{ladder}}$); and (c) whether the grapes are *reachable* or not (i.e., $\overline{\textit{reachable}}$). These propositional variables are linked via the **axioms** in (2.13):

$$\begin{aligned} &\textit{low} \rightarrow \textit{reachable} \\ &(\overline{\textit{low}} \wedge \overline{\textit{ladder}}) \rightarrow \overline{\textit{reachable}} \\ &(\overline{\textit{low}} \wedge \textit{ladder}) \rightarrow \textit{reachable} \end{aligned} \tag{2.13}$$

We formalize this story as follows. In the initial situation (i.e., before the fox's first attempt to get the grapes) the CP-net is:

$$CPT(G) = \textit{grapes} \succ \overline{\textit{grapes}} \qquad \textbf{(intrinsic preference)}$$

$$\begin{aligned} CPT(A) = {} &\textit{grapes} \wedge \textit{reachable} : \textit{get} \succ \textit{walk} \\ &\textit{grapes} \wedge \overline{\textit{reachable}} : \textit{walk} \succ \textit{get} \\ &\overline{\textit{grapes}} : \textit{walk} \succ \textit{get} \end{aligned}$$

Since in this situation there is no ladder ($\overline{\textit{ladder}}$), the fox learns from his attempt to *get* the grapes with no success that they are not reachable. He now believes $\overline{\textit{reachable}}$ and hence knows via the axioms (2.13) $\overline{\textit{low}}$. When believing $\overline{\textit{reachable}}$, the fox chooses the action to *walk* away.

In the second situation where the fox spots a ladder after his first failed attempt to get the grapes, there are two cases: 1) one in which he takes the decision to try again; and 2) one in which he takes the decision to give up.

For 1), the CP-net is the same as in the first situation. This is similar to Spohn's example of extrinsic preference change: the change is to the fox's *decision*, thanks to a change in his beliefs, and not to his preferences. From his first attempt, the fox knows $\overline{\textit{low}}$ and thanks to the *ladder*, the fox changes his beliefs via the axioms (2.13) to *reachable*. So he decides to *get* the grapes. Therefore, there is no (intrinsic) preference change; but rather a change to decision making that's afforded by different beliefs.

For 2), since the fox spots the ladder but does not try again to get the grapes, he must have changed his preferences for the grapes (unless he revised his domain-level axiom that the ladder makes them reachable). He confirms this by saying the grapes are sour. By Definition 5, this triggers a revision to the initial CP-net, resulting in the following:

$$CPT(G) = \overline{grapes} \succ grapes \qquad \textbf{(intrinsic preference)}$$

$$\begin{aligned} CPT(A) = {} & \overline{grapes} : walk \succ get \\ & grapes \wedge reachable : get \succ walk \\ & grapes \wedge \overline{reachable} : walk \succ get \end{aligned}$$

2.4 Conclusion

In this chapter, we described several decision and game problems for which a static and pre-defined utility function isn't sufficient for reasoning about rational action. First, there are practical problems in games where all possible ways of getting from the current state to an end state aren't surveyable. Static, pre-defined preferences are also untenable on conceptual grounds. One can be playing a game while at the same time discovering exactly what the game is—discovering new possible states or actions that require a changing not only one's belief model but also one's preference model. Secondly, one can simply change one's preferences in the absence of any belief change or new discovery, for instance through a change in taste. We agreed with Spohn (2009) that modelling such scenarios using global decision models models doesn't provide sufficiently rich information for discriminating among the intuitively compelling optimal strategies.

We proposed to model preference change with CP-nets, a compact and qualitative representation of preferences. We defined preference update and showed that it preserves consistency. The explicit encoding in CP-nets of dependencies among preferences guides the revision process: the agent aims to minimise the overall number of existing preferences that he removes, and he is more reluctant to remove preferences X that influence other preferences Y than he is to remove Y. The result is a model of preference revision similar to an AGM model of belief revision (Alchourrón et al., 1985). We applied this model to the motivating examples from Section 2.1. We showed how it complements earlier research about preference change and in particular we demonstrated how it overcomes Spohn's (2009) criticisms of global models of dynamic preferences.

While the study presented here concerns preferences in general, in the following chapters we focus on preferences as they are verbalised in dialogues and Chapter 5 will complement the current study by providing more precise rules to model preference changes as they are verbalised.

Chapter 3

Data and preference annotation

3.1 Presentation of each corpus genre **78**

3.1.1 ***Verbmobil*** and ***Booking*** corpora: cooperative negotiations 78

3.1.2 ***Settlers*** corpus: a competitive game 83

3.2 Preference annotation **89**

3.2.1 How are preferences linguistically expressed? 89

3.2.2 Preference annotation scheme for cooperative dialogues (***Verbmobil*** and ***Booking*** corpora) 91

3.2.3 Preference annotation scheme for non-cooperative dialogues (***Settlers*** corpus) 99

3.3 Conclusion **107**

Previous chapters were dedicated to a theoretical study of preferences while the following ones describe more empirical work about how to linguistically extract preferences and reason in negotiation and bargaining dialogues. In the current chapter, we first present the different corpus genres we worked on. We then provide a complete study of how preferences are expressed in these corpora and propose a new preference annotation scheme.

In Section 3.1, we present each corpus genre and their given annotation. First, we study two corpora where agents negotiate about how to carry out a common goal. For ***Verbmobil***, the goal is to find a meeting time, and for ***Booking*** it is to arrange a reservation. Then, we study the ***Settlers*** corpus, a corpus of on line chats concerning the non-cooperative bargaining game *The Settlers of Catan*. We also detail the already existing annotations provided for each corpus. For ***Verbmobil*** and ***Booking***, there is a discursive annotation which segments the dialogues into segments called *elementary discourse units* and links them together with *rhetorical relations*. The dialogue annotation scheme for ***Settlers*** is multi-layered. It includes a discursive annotation as for ***Verbmobil*** and ***Booking***. It also provides some information about more strategic aspect of the game like dialogue acts that are specific to bargaining (*offers*, *counteroffers*, etc.) and information about the *givable* and/or *receivable* resources that each segment expresses.

In Section 3.2, we study how preferences are linguistically expressed in these corpus. We first describe the annotation methodology and detail the inter-annotator agreement study on the cooperatives dialogues from ***Verbmobil*** and ***Booking***. We then expose how to extend this work to annotate the more complex data from our ***Settlers*** corpus and present the new inter-annotator agreements. Our results show that preferences can be easily annotated by humans.

3.1 Presentation of each corpus genre

3.1.1 *Verbmobil* and *Booking* corpora: cooperative negotiations

3.1.1.1 Presentation of the dialogues

Our data come from two corpora: one already-existing, ***Verbmobil***, and one built in-house, ***Booking***.

The first corpus is composed of 35 dialogues randomly chosen from the existing corpus Verbmobil (Wahlster, 2000), where two agents discuss on when and where to set up a meeting. Here is a typical fragment:

(3.1)
π_1 A: Shall we meet sometime in the next week?
π_2 A: What days are good for you?
π_3 B: I have some free time on almost every day except Fridays.
π_4 B: Fridays are bad.
π_5 B: In fact, I'm busy on Thursday too.
π_6 A: Next week I am out of town Tuesday, Wednesday and Thursday.
π_7 A: So perhaps Monday?

The second corpus was built from various English language learning resources, available on the Web.[1] It contains 21 randomly selected dialogues, in which one

[1]e.g., http://www.bbc.co.uk/worldservice/learningenglish/

agent (the customer) calls a service to book a room, a flight, a taxi, etc. Here is a typical fragment:

(3.2)
π_1 A: Northwind Airways, good morning. May I help you?
π_2 B: Yes, do you have any flights to Sydney next Tuesday?
π_3 A: Yes, there's a flight at 16:45 and one at 18:00.
π_4 A: Economy, business class or first class ticket?
π_5 B: Economy, please.

In ***Verbmobil***, agents are cooperative because they both want to satisfy a common goal: that is, to fix a meeting. Where they may disagree, is about when and where to fix the meeting. So during the negotiation, they express their preferences about days, hours and places in order to fix a meeting at their best time. In this corpus, most of the preferences concerns a vocabulary set of days and times.

Booking is also a corpus of cooperative negotiations where the agents share a common goal. In this corpus, the consumer wants to make a booking and the seller wants to satisfy his constraints. In this corpus, the vocabulary set is larger than for ***Verbmobil***. Again preferences concerns days and times about when to fix the booking but also it concerns more diverse outcomes specific to each booking like the class (economy, business or first class) for a flight, the size of the room (single or double) for an hotel.

3.1.1.2 The discourse-level annotation

To represent the discourse context, we use the Segmented Discourse Representation Theory, SDRT (Asher and Lascarides, 2003). It structures discourse into elementary discourse units (EDUs) that are linked together by rhetorical relations. The segments cover a single clause of speech or complex segments (CDUs) composed of other segments and their relations. The rhetorical relations used for the annotation are presented below.

- ***Elaboration(***π_1***,*** π_2***)*** ($Elab$) relates EDUs or CDUs whenever the second unit π_2 provides more information about the eventuality introduced in the first constituent π_1. This relation can be further specialised as follows:
  - ***Plan-Elaboration(***π_1***,*** π_2***)*** (P-$Elab$) is an *Elaboration* where π_2 elaborates the plan expressed in π_1.

  - ***Question-Elaboration(**π_1**,** π_2**)*** (Q-$Elab$) is a relation that is just like P-$Elab$, except that the second argument to the relation (π_2) labels a question rather than a proposition.

- ***Correction(**π_1**,** π_2**)*** ($Corr$) holds when the second unit π_2 corrects what was said in the first one π_1. This relation can be further specialised as follows:

  - ***Plan-Correction(**π_1**,** π_2**)*** (P-$Corr$) means that π_2 expresses intentions or goals that are incompatible with the ones expressed in π_1.
  - ***Question-Correction(**π_1**,** π_2**)*** (Q-$Corr$) is a *Correction* where the second argument π_2 labels a question rather than a proposition.

- ***Explanation(**π_1**,** π_2**) (**$Expl$**)*** holds when π_2 explains why, or gives the cause of, what happened in π_1. Specific kinds of *Explanation* are:

  - ***IExplanation(**π_1**,** π_2**)*** is an *Intentional Explanation*. It holds when π_2 expresses a reason for doing what is said in π_1, i.e. in π_1: *He took is girlfriend in a fancy restaurant* π_2: *to impress her*.
  - ***Explanation*(**π_1**,** π_2**)*** ($Expl*$) is a meta-talk relation. It holds when π_2 explains why the agent said, asked or requested π_1, i.e. in π_1: *Could you close the windows ?* π_2: *I'm cold.*

- ***Question Answer Pair(**π_1**,** π_2**)*** (QAP) is used to link an answer (π_2) to the question (π_1) it is an answer to.

- ***Clarification Question(**π_1**,** π_2**)*** (Q-$Clar$) holds when π_2 is a question that ask for a clarification (or justification) of what was said in π_1.

- ***Comment(**π_1**,** π_2**)*** holds if π_2 provides an opinion or evaluation of the content associated with π_1.

- ***Narration(**π_1**,** π_2**)*** holds when the main eventualities of the first unit π_1 and the second unit π_2 occur in sequence.

- ***Continuation(**π_1**,** π_2**)*** is like *Narration*, except there is no sequence of actions. *Continuation* often holds between two EDUs or CDUs when they both elaborate or provide background to the same segment.

  - ***Question-Continuation(**π_1**,** π_2**)*** (Q-$Cont$) is a *Continuation* where π_2 is a question.

- ***Contrast(**π_1**,** π_2**)*** holds when π_1 and π_2 have similar semantic structures, but contrasting themes, i.e. sentence topics, or when one constituent negates a default consequence of the other.

- ***Parallel(π_1, π_2)*** has the same structural requirements as *Contrast*, but instead requires π_1 and π_2 to share a common theme.
- ***Result(π_1, π_2)*** connects a cause to its effect, i.e. the main eventuality of the first argument π_1 is understood to cause the eventuality given by π_2.
- ***Conditional(π_1, π_2)*** marks the presence of a conditional between two clauses where the first discourse unit π_1 is a hypothesis while the second π_2 is a consequence of the hypothesis.
- ***Alternation(π_1, π_2)*** marks the presence of a disjunction between two clauses.
- ***Summary(π_1, π_2)*** holds when π_2 summarises what was said in π_1.
- ***Acknowledgement(π_1, π_2)*** ($Ackn$) holds when π_2 acknowledges the receipt of information in π_1. Acknowledgments can function at different levels, for example an agent can say "OK" just to signal to his interlocutor that he understood what was said. But he can also says "OK" if he understands and accepts what his interlocutor has said.

For ***Verbmobil***, the discourse annotation is given by Baldridge and Lascarides (2005a). For ***Booking***, annotation was made by consensus between two annotators using the same set of rhetorical relations used to annotate ***Verbmobil***. To illustrate this annotation, consider again the ***Verbmobil*** example (3.1). The corresponding discourse structure is given in Figure 3.1.

Intuitively, A's question π_1 reveals his preference for meeting next week and *Q-Elab*(π_1, π_2) entails that any answer to π_2 must elaborate a plan to achieve the preference revealed by π_1; this makes π_2 paraphrasable as "What days next week are good for you?", which does not add new preferences. Nevertheless, B's response in π_3 to π_5 to A's elaborating question π_2 reveals that he has adopted A's preference. In effect, A's preference is adopted in π_3, which specifies a non-empty extension for what days to meet. Inferences about B's preferences evolve as he gives his extended answer: from π_3 alone one would infer a preference for meeting any day next week other than Friday and its explanation π_4 would maintain this. But the correction π_5 compels A to revise his inferences about B's preference for meeting on Thursday. These inferences about preferences arise from both the content of B's utterances and the semantic relations that connect them together. A's response π_6 reveals that he disprefers Tuesday, Wednesday and Thursday, thereby refining the preferences that he revealed last time he spoke. A's follow-up proposal π_7 then reinforces the inference from π_6 that among Monday, Tuesday and Wednesday – the days that B prefers, A prefers Monday. This may not match his preferred day when the dialogue started: perhaps that was Friday. Further dialogue may compel agents to revise their preferences as they learn about the domain and about each other.

This example shows that agents' preferences depend upon the compositional interpretation of the discourse structure over EDUs. The constraints are different for different discourse relations, reflecting the fact that the semantics of connections between EDUs influences how their preferences relate to one another.

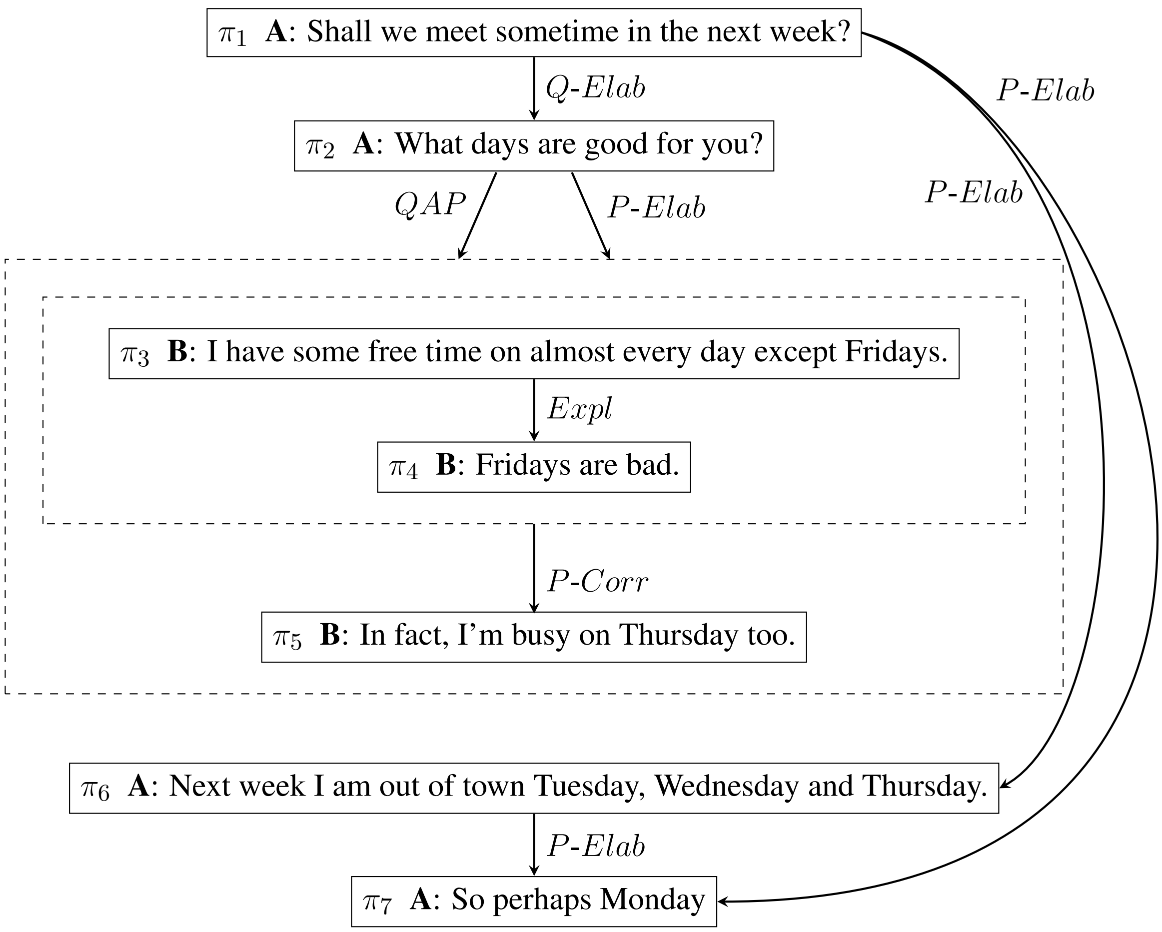


Figure 3.1: The discourse structure for a ***Verbmobil*** dialogue.

3.1.2 *Settlers* corpus: a competitive game

3.1.2.1 Presentation of the game

Our data come from on line chats concerning the game *The Settlers of Catan*, a competitive win-lose game that involves negotiations. In this corpus, humans are playing an online version of the game (Afantenos et al., 2012b) where players must converse in a chat interface to carry out trades and the state of the game is recorded and aligned with players' conversations (see Figure 3.2 for an illustration). The chats involve principally bargaining over resources but contain also comments about strategic aspects of the game. Each game contains several dozen self-contained bargaining dialogues.

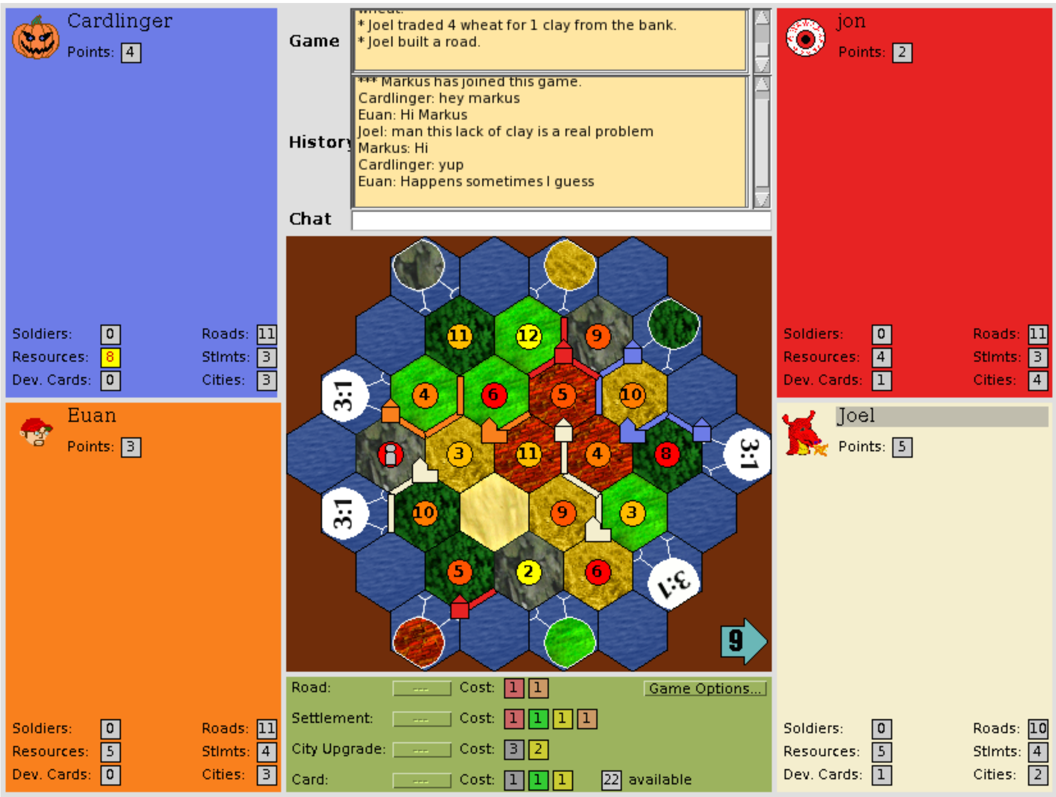


Figure 3.2: *The Settlers of Catan* chat interface.

Two to four players build settlements and cities connected by roads on the island of Catan. They must use certain resources (clay, ore, sheep, wheat and wood) in different combinations to build their constructions: a road requires 1 clay and 1 wood; a settlement requires 1 wood, 1 clay, 1 wheat and 1 sheep; and a city requires 2 wheats and 3 ores. Victory points get awarded to players in several ways: e.g., by building a settlement (1 point) or a city (2 points). It is a win lose game: the first player with 10 victory points wins. Players acquire resources in several ways, in particular through agreed trades with other players. Since players can't recall all the trades occurred during the game and since some methods (e.g., robbing) are hidden from view, players lack complete information about their opponents' resources.[2]

Unlike ***Verbmobil*** and ***Booking***, where the agents share a common goal (to fix a meeting or to make a booking) and want the negotiation to succeed, in ***Settlers*** the negotiation is more conflictual since the agents are not always interested in a successful negotiation. Indeed, negotiations helps the players to obtain resources that bring victory. However, since *The Settlers of Catan* is a win-lose game, players can't win together and do not wish to satisfy negotiations that would be too beneficial for their opponents. Here is a typical negotiation dialogue from our corpus:

(3.3) π_1 $Euan$: And I alt tab back from the tutorial. What's up?

π_2 $Joel$: do you want to trade?

π_3 $Card.$: joel fancies a bit of your clay

π_4 $Joel$: yes

π_5 $Joel$: !

π_6 $Euan$: Whatcha got?

π_7 $Joel$: wheat

π_8 $Euan$: I can wheat for 1 clay or 1 wood.

π_9 $Joel$: awesome

This negotiation dialogue is typical; it involves some creative vocabulary (*alt tab* as a lexical verb) or verb ellipsis without a surface antecedent (*I can wheat for clay*), with imperfect knowledge/recall amply evident (Euan's *what's up?*). There are also strategic comments (π_3) and underspecified bargaining moves (as π_2 and π_7) that get specified as more information becomes common knowledge.

Most of the turns in the chats involve negotiation and represent offers, counteroffers, and acceptances or rejections of offers. All of these convey preferences. For instance:

(3.4) Anybody have any sheep for wheat?

[2]See http://www.catan.com/ for a more detailed description of the rules.

This dialogue move concerns several preferences. First, it conveys a preference to trade with someone unspecified; offers to trade themselves are complex preferences involving a preference for some amount of sheep over alternatives and given that the agent receives sheep, she is willing to give away some of her wheat over the alternatives. Crucially, it does not convey a preference to give away wheat in a context where she receives nothing or something other than sheep.

Note that here preferences are *underspecified*, and the offer is also underspecified. In line with a non-cooperative bargaining game, the preferences and offers that a speaker reveals via a dialogue move are less specific than an executable trade requires, where the trading partners and the type and quantity of the resources they exchange must all be defined. We interpret (3.4) as a pre-negotiation question or signal in some form of sender-receiver game. Such general moves are essentially information seeking moves, giving evidence that humans playing *The Settlers of Catan* are operating with imperfect information with, in particular, incomplete information about their opponents' preferences. Our agents' cognitive limitations force them to rely on local scoring functions, thus allowing for Pareto-improving deals in the overall context of a conflicting strategic situation. The fact that our agents also play with imperfect information of several kinds and the fact that communication need not be credible might threaten the existence of strategically stable deals. In fact, many offers to trade result in no trade being agreed and executed. While observed negotiation failure would be puzzling in a bargaining game with perfect information, it occurs relatively frequently in the ***Settlers*** corpus.

3.1.2.2 The strategic annotation

Each turn logs what a player enters in the chat window and also aspects of the game state at the time: his resources, the state of the game board and a time stamp. An automatic pre-annotation segments the dialogue into turns and the author of each turn is automatically given. Then, annotators provide the discourse structure of the dialogue where each turn is further segmented into Elementary Discourse Units (EDUs). The annotators then, have to specify the strategic annotation of each EDU with the associated dialogue act (*offers*, *counteroffers*, etc.) and information about the *givable* and/or *receivable* resources expressed in the segment.

As for ***Verbmobil*** and ***Booking***, the discourse annotation provides the structure of the dialogues according to the Segmented Discourse Representation Theory, SDRT. The annotation is established by consensus between at least two annotators. Figure 3.3 shows the discourse structure corresponding to example (3.3).

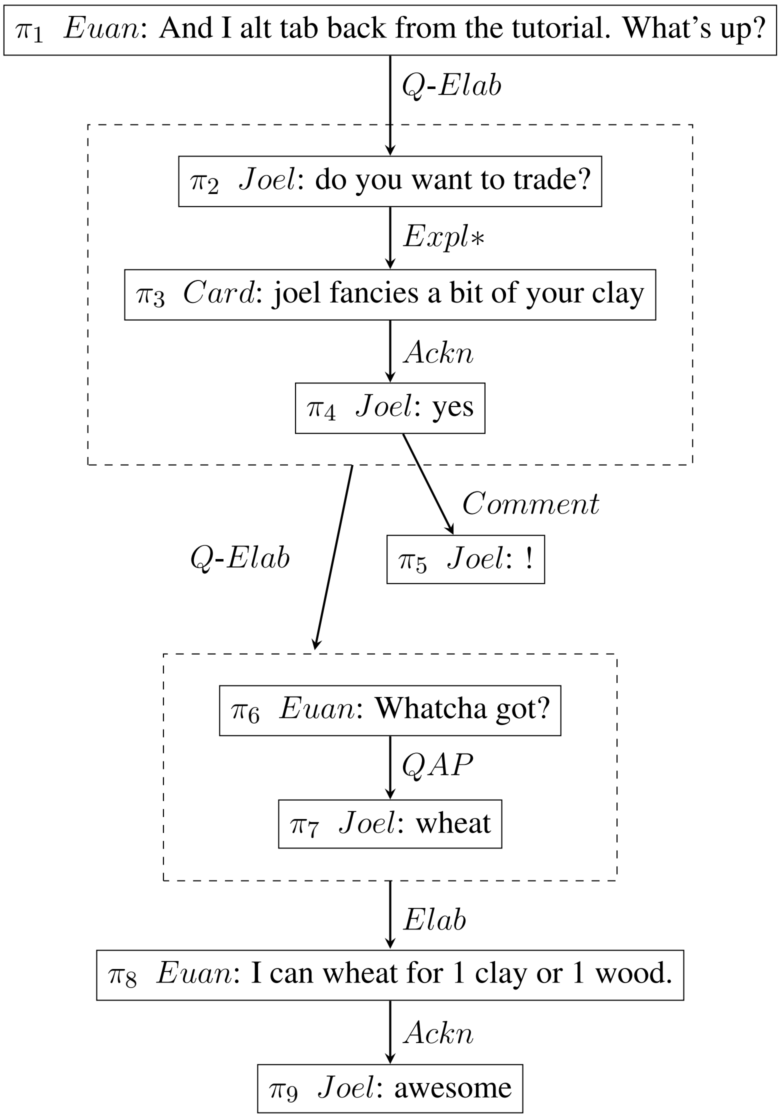


Figure 3.3: The discourse structure for a ***Settlers*** dialogue.

The strategic annotation scheme includes: (1) the addressee of the turn; (2) a characterization of each EDU in terms of a basic theory of speech acts (assertion, question, request) as well as dialogue acts that are specific to bargaining (offers, counteroffers, etc.) and (3) associated information about the givable and/or receivable resources that offers, counteroffers, etc. express.

Two annotators received training on 77 dialogues, totalling 699 EDUs. They then both annotated the remaining dialogues independently (2741 EDUs and 511 dialogues in total) using the GLOZZ annotation platform[3]. The kappas for inter-annotator agreement are given below (see appendix A.2 for an explanation of kappa calculus).

Table 3.1 shows the annotation for example (3.3). A dialogue turn can express an offer, a counteroffer, an acceptance or rejection of offers, or a commentary on the above or on moves in the game. All except the last provide clues about preferences: e.g., which players a speaker wants to execute a trade with; or what resources to exchange. For instance, the utterance *I can wheat for clay* concerns several preferences. First, it conveys the speaker's preference to acquire some wheat over alternatives and to give some clay in return. Some turn provides information about unacceptable outcomes as in the Refusal *I'm fine for wheat at the mo* where we learn that receiving wheat is currently not an acceptable for the speaker.

ID	Dialogue Act	Text	Speaker	Addressee	Resource
π_1	Other	And I alt tab back from the tutorial. What's up?	Euan	All	
π_2	Offer	do you want to trade?	Joel	Euan	
π_3	Other	joel fancies a bit of your clay	Card.	Euan	
π_4	Other	yes	Joel	Euan	
π_5	Other	!	Joel	Euan	
π_6	Counteroffer	Whatcha got?	Euan	Joel	
π_7	Counteroffer	wheat	Joel	Euan	Givable (wheat, ?)
π_8	Counteroffer	I can wheat for clay.	Euan	Joel	Receivable (wheat, 1) Givable (clay, 1)
π_9	Accept	awesome	Joel	Euan	

Table 3.1: The strategic annotation for a ***Settlers*** dialogue.

[3] http://www.glozz.org/

Speech act and dialogue act annotation.

First, the annotators have to specify the *addressee* of each EDU which may be the whole group of players, any subset thereof or another individual. Sometimes it may not be clear who's being addressed, in which case this field is completed with a ? mark.

Each EDU also has a *speech act*: it's either a *Question* (e.g., *Do you want to trade?*), a *Request* (e.g., *Give me two sheep*) or an *Assertion* (e.g., *I need sheep*).

The annotators also specify the *dialogue act* of each EDU: *Offer*, *Counteroffer* (which may be a reply to an offer or a specification of an offer the speaker has already made), *Accept* or *Refusal* (of an offer addressed to the emitter), and *Other*. *Other* labels units that either comment on strategic moves in the game or are not directly pertinent to bargaining as shown in EDU π_1 in Table 3.1. For dialogue act annotation, we got a kappa of 0.79.

Resource type annotation.

Inside each EDU associated with its dialogue act, we can have "Resource" units, which identify resource-denoting words (as "clay", "ore", etc.). Annotators then specify an associated *feature structure*, which captures (partial) information that the EDU expresses about the *type* and *quantity* of resources (see the Resource column in Table 3.1 where the format is $Type(resource\ word,\ quantity)$). The type is either *Givable*, *Not Givable*, *Receivable* or *Not Receivable*. The annotation also includes resources that are *Possessed* and *Not Possessed* but we don't use these here.

The structure can take Boolean combinations of resources via two operators AND and OR, that respectively stand for conjunction (the agent expresses two preferences and he prefers to achieve one of them if he cannot have both, such as in ***I need clay and wood*** annotated as $Receivable(clay, ?)AND(wood, ?)$) and disjunction (free choice) of preferences (e.g., *I can give you clay or wood* annotated as $Givable(clay, ?)OR(wood, ?)$).

We allow attributes to have unknown values: the annotation tool inserts a ? in these cases.

We also insist that the annotators resolve anaphoric dependencies when specifying values to attributes, as shown in pi_2 in example below with the following resource annotation: $Not\ Givable(Anaphoric, ?)\ Anaphora\ Link : (mine{,}clay)$.

The inter-annotator agreements on the kind of the resources is 0.80.

π_1 A: i need clay, any1 have?

π_2 B: need mine sorry

3.2 Preference annotation

We showed in the previous section (see 3.1.1.2) how the agents' preferences depend upon the compositional interpretation of the discourse structure over elementary discourse units (EDUs). In this section, we analyze how the outcomes and the dependencies between them are linguistically expressed and propose a new scheme to annotate the preferences expressed in each segment. Two annotators were involved in this process.

3.2.1 How are preferences linguistically expressed?

To analyze how preferences are linguistically expressed in each EDU, we must: (1) identify the set Ω of outcomes, on which the agent's preferences are expressed, and (2) identify the dependencies between the elements of Ω by using a set of specific operators, i.e. identify the agent's preferences on the stated outcomes. Consider the segment "Let's meet Thursday or Friday". We have Ω = {*meet Thursday*, *meet Friday*} where outcomes are linked by a disjunction that means the agent is ready to act for one of these outcomes, preferring them equally. It is this dependency that allows us to infer the preference relation between each couple of outcomes and thus, to identify, given two outcomes, o_1 and $o_2 \in O$, the preference relation between them (i.e. o_1 is preferred to o_2, o_2 is preferred to o_1 or o_1 is indifferent to o_2). Within an EDU, preferences can be expressed in different ways. They can be atomic preference statements or complex preference statements.

Atomic preferences. Atomic preference statements are of the form "I prefer o_1", "Let's o_1", or "We need o_1", where o_1 describes an outcome. o_1 may be a definite noun phrase ("Monday", "next week", "almost every day"), a prepositional phrase ("at my office") or a verb phrase ("to meet").

Preferences can be expressed within comparatives and/or superlatives ("a cheaper room" or "the cheapest flight").

Preferences can also be expressed in an indirect way using questions. Although not all questions entail that their author commits to a preference, in many cases they do. That is, if A asks "can we meet next week?" he implicates a preference for meeting. For negative and wh-interrogatives, the implication is even stronger.

Expressions of opinions, sentiment or politeness can also be used to indirectly introduce preferences. For example, in ***Verbmobil***, the utterance "Tuesday would be good" introduce a preference for meeting on Tuesday. In ***Booking***, the segment "economy please" indicates the agent's preference to be in an economy class. Preferences can also be introduced by specific words of preferences, such as verbs (*to prefer, to favour*, etc.) or adjectives (*favourite, preferable*, etc.).

EDUs can also express preferences via modalities; "Thursday would be good" or "I can meet on Thursday" tells us that Thursday is a possible day to meet, it is an acceptable outcome.

A negative preference expresses an unacceptable outcome, i.e. what the agent does not prefer. Negative preference can be expressed explicitly with negation words ("I don't want to meet on Friday") or inferred from the context ("I am busy on Monday").

Complex preferences. Preference statements can also be complex, expressing dependencies between outcomes. Borrowing from the language of *conditional preference networks* or CP-nets (Boutilier et al., 2004), we recognize that some preferences may depend on another action. For instance, given that I have chosen to eat fish, I will prefer to have white wine over red wine—something which we express as *eat fish* : *drink white wine* $\succ$ *drink red wine*.

Among the possible combinations, we find conjunctions, disjunctions and conditionals.

With conjunctions of preferences, as in "Could I have a breakfast and a vegetarian meal?" or in "Mondays and Fridays are not good?", the agent expresses two preferences (respectively over the acceptable outcomes breakfast and vegetarian meal and the non acceptable outcomes not Mondays and not Fridays) that he wants to satisfy and he prefers to have one of them if he can not have both. Hence a conjunction between outcomes o_1 and o_2 means $o_1 \succ \overline{o_1}$ and $o_2 \succ \overline{o_2}$.

The semantics of a disjunctive preference is a free choice one. For example in "either Monday or Tuesday is fine for me" or in "I am free Monday and Tuesday", the agent states that either Monday or Tuesday is an acceptable outcome and he is indifferent between the choice of the outcomes. Hence a disjunction between outcomes o_1 and o_2 means $o_2 : o_1 \sim \overline{o_1}, \overline{o_2} : o_1 \succ \overline{o_1}$ and $o_1 : o_2 \sim \overline{o_2}, \overline{o_1} : o_2 \succ \overline{o_2}$.

Finally, some EDUs express conditional among preferences. For example, in the sentence "What about Monday, in the afternoon?", there are two preferences: one for the day Monday, and, given the Monday preference, one for the time afternoon (of Monday), at least for one syntactic reading of the utterance. Hence a conditional dependency between outcomes o_1 and o_2 means $o_1 \succ \overline{o_1}$ and $o_1 : o_2 \succ \overline{o_2}$.

3.2.2 Preference annotation scheme for cooperative dialogues (*Verbmobil* and *Booking* corpora)

3.2.2.1 Preference formalisation

Suppose a language with non-boolean operators taking outcome expressions as arguments. For representing negative preference, we use the unary operator *not*. For representing complex preferences, we use the binary operators $\&$, $\bigtriangledown$ and $\mapsto$ that represent respectively conjunctions, disjunctions and conditionals. While the logical form of an atomic preference statement is something of the form Pref(o_1), we abbreviate this in the annotation language, using just the outcome expression o_1 to denote that the agent prefers o_1 to the alternatives, i.e. $o_1 \succ \overline{o_1}$. In our ***Verbmobil*** annotation, o_1 is typically a Noun Phrase (NP) denoting a time or place; o_1 as an outcome is thus shorthand for *meet on* o_1 or *meet at* o_1. For ***Booking***, o_1 is short for *reserve* or *book* o_1.

We give below an example of how some EDUs are annotated. <o>_i indicates that o is the outcome number i in the EDU, the symbol // is used to separate the two annotation levels and brackets indicate how outcomes are attached.

(3.5)

π_1 <Tuesday the sixteenth>_1 I got class <from nine to twelve>_2? // 1 $\mapsto$ not 2

π_2 What about <Friday afternoon>_1, <at two thirty>_2 or <three>_3, // 1 $\mapsto$ (2 $\bigtriangledown$ 3)

π_3 <The room with balcony>_1 should be equipped <with a queen size bed>_2, <the other one>_3 <with twin beds>_4, please. // (1 $\mapsto$ 2) $\&$ (3 $\mapsto$ 4)

In π_1, the annotation tells us that we have two outcomes and that the agent prefers outcome 1 over any other alternatives and given that, he does not prefer outcome 2.

In π_2, the annotation tells us that the agent prefers to have one of outcome 2 and outcome 3 satisfied given that he prefers outcome 1. In this example, the free choice between outcome 2 and outcome 3 is lexicalized by the coordinating conjunction “or”.

On the contrary, π_3 is a more complex example where there is no discursive marker to find that the preference operator between the couples of outcomes 1 and 2 on one hand, and 3 and 4 on the other hand, is the conjunctive operator $\&$.

3.2.2.2 Inter-annotator agreements

Our two corpora (***Verbmobil*** and ***Booking***) were annotated by two annotators using the previously described annotation scheme. We performed an intermediate analysis of agreement and disagreement between the two annotators on two ***Verbmobil*** dialogues. Annotators were thus trained only for ***Verbmobil***. The aim is to study to what extent our annotation scheme is genre dependent. The training allowed each annotator to understand the reason of some annotation choices. After this step, the dialogues of our corpora have been annotated separately, discarding those two dialogues. Table 3.2 presents some statistics about the annotated data in the gold standard.

	Verbmobil	***Booking***
No. of dialogues	35	21
No. of outcomes	1081	275
No. of EDUs with outcomes	776	182
% with 1 outcome	71%	70%
% with 2 outcomes	22%	19%
% with 3 or more outcomes	8%	11%
No. of unacceptable outcomes (not)	266	9
No. of conjunctions (&)	56	31
No. of disjunctions ($\bigtriangledown$)	75	29
No. of conditionals ($\mapsto$)	184	37

Table 3.2: Statistics for the preference annotation in ***Verbmobil*** and ***Booking***.

We computed four inter-annotator agreements on: (a) outcome identification, (b) outcome acceptance, (c) outcome attachment and finally (d) operator identification. Table 3.3 summarises our results.

Agreements on outcome identification. Two inter-annotator agreements were computed using Cohen's kappa (see appendix A.2). One based on an *exact* matching between two outcome annotations (i.e. their corresponding text spans), and the other based on a *lenient* match between annotations (i.e. there is an overlap between their text spans as in "2p.m" and "around 2p.m"). We obtained an exact agreement of 0.66 and a lenient agreement of 0.85 for both corpus genres.

	Verbmobil	***Booking***
Outcome identification (kappa)	exact: 0.66 lenient: 0.85	
Outcome acceptance (kappa)	0.90	0.95
Outcome attachment (F-measure)	93%	82%
Operator identification (kappa)	0.93	0.75

Table 3.3: Inter-annotator agreements for ***Verbmobil*** and ***Booking***.

We made the gold standard after discussing cases of disagreement. We observed four cases. The first one concerns redundant preferences which we decided not to keep in the gold standard. In such cases, (where we usually found relations like *Result*, *Summary*, *Q-Continuation*), the second EDU π_2 does not introduce a new preference, neither does it correct the preferences stated in π_1; rather, the agent just wants to insist by repeating already stated preferences, as in the following example, where we have *Result*(π_1, π_2):

π_1 A: Thursday, Friday, and Saturday I am out.

π_2 A: So those days are all out for me,

The second case of disagreement comes from anaphora which are often used to introduce new, to make more precise or to accept preferences. Hence, we decided to annotate them in the gold standard. Here is an example, where we have *P-Elab*(π_1, π_2):

π_1 A: One p.m. on the seventeenth?

π_2 B: That sounds fantastic.

The third case of disagreement concerns preference explanation. We chose not to annotate these expressions in the gold standard because they are used to explain already stated preferences. In the following example, where we have *Explanation*(π_1, π_2), one judge annotated “from nine to twelve” to be expressions of preferences while the other did not:

π_1 A: Monday is really not good,

π_2 A: I have got class from nine to twelve.

Finally, the last case of disagreement comes from preferences that are not directly related to the action of fixing a date to meet but to other actions, such as having lunch, choosing a place to meet, etc. Even though those preferences were often missed by annotators, we decided to keep them, when relevant.

Agreements on outcome acceptance. The aim here is to compute the agreement on the *not* operator, that is if an outcome is acceptable, as in "<Mondays>_1 are good // 1", or unacceptable, as in "<Mondays>_1 are not good // not 1". We got a Cohen's kappa of 0.9 for ***Verbmobil*** and 0.95 for ***Booking***.

The main case of disagreement concerns anaphoric negations that are inferred from the context, as in π_2 below where annotators sometimes fail to consider "in the morning" as unacceptable outcome:

π_1 A: Tuesday is kind of out,

π_2 A: Same reason in the morning

Same case of disagreement in this example where "Monday" is an unacceptable outcome:

π_1 well, I am, busy in the afternoon of the twenty sixth,

π_2 that is Monday

Agreements on outcome attachment. Since this task involves structure building, we computed the agreement using the F-score measure. The agreement was computed on the previously built gold standard once annotators discussed cases of outcome identification disagreements. We compared how each outcome is attached to the others within the same EDU. This agreement concerns EDUs that contain at least three outcomes, that is 8% of EDUs from ***Verbmobil*** and 11% of EDUs from ***Booking***. When comparing annotations for the example π_1 below, there is three errors, one for outcome 2, one for 3 and one for 4.

π_1 <for the next week>_1 the only days I have open are <Monday>_2 or <Tuesday>_3 <in the morning>_4.

* Annotation 1: $1 \mapsto (2 \bigtriangledown (3 \mapsto 4))$
* Annotation 2: $1 \mapsto ((2 \bigtriangledown 3) \mapsto 4)$

π_2 Total time is <15 hours>_1, <XYZ>_2 takes <11 hours>_3. // $1 \mapsto (2 \mapsto 3)$

* Annotation 1: $1 \mapsto (2 \mapsto 3)$
* Annotation 2: $1 \bigtriangledown (2 \mapsto 3)$

We obtained an agreement of 93% for ***Verbmobil*** and 82% for ***Booking***.

Agreements on outcome dependencies. Finally, we computed the agreements for each couple of outcomes on which annotators agreed about how they are attached.

In ***Verbmobil***, the most frequently used binary operator is $\mapsto$. Because the main purpose of the agents in this corpus is to schedule an appointment, the preferences expressed by the agents are mainly focused on concepts of time and there are many conditional preferences since it is common that preferences on specific concepts depend on more broad temporal concepts. For example, preferences on hours are generally conditional on preferences on days. In ***Booking***, there are almost as many $\&$ as $\mapsto$ because independent and dependent preferences are more balanced in this corpus. The agents discuss preferences about various criteria that are independent. For example, to book a hotel, the agent express his preferences towards the size of the bed (single or double), the quality of the room (smoker or nonsmoker), the presence of certain conveniences (TV, bathtub), the possibility to have breakfast in his room, etc. Within an EDU, such preferences are often expressed in different sentences (compared to ***Verbmobil*** where segments' lengths are smaller) which lead annotators to link those preferences with the operator $\&$. Conditionals between preferences hold when decision criteria are dependent. For example, the preference for having a vegetarian meal is conditional on the preference for having lunch. There also are conditionals between temporal concepts, for example, to choose the time of a flight. In one segment, the preferences on such independent features are linked with the operator $\mapsto$.

Table 3.4 shows the kappa for each operator on each corpus genre. The kappa, averaged over all the operators, is 0.93 for ***Verbmobil*** and 0.75 for ***Booking***. We observe two main cases of disagreement: between $\bigtriangledown$ and $\&$, and between $\&$ and $\mapsto$. These cases are more frequent for ***Booking*** mainly because annotators were not trained on this corpus and may explain why the kappa is lower than for ***Verbmobil***. We discuss below the main two cases of disagreement.

Confusion between $\bigtriangledown$ and $\&$. The same linguistic realizations do not always lead to the same operator. For instance, in "<Monday>_1 and <Wednesday>_2 are good" we have $1 \bigtriangledown 2$ whereas in "<Monday>_1 and <Wednesday>_2 are not good" or in "I would like a <single room>_1 and a <taxi>_2" we have respectively $not\ 1\ \&\ not\ 2$ and $1\ \&\ 2$. In section 3.2.1, we shown that some linguistic realizations

	Verbmobil	***Booking***
&	0.90	0.66
$\triangledown$	0.97	0.89
$\mapsto$	0.92	0.71

Table 3.4: Agreements on binary operators for ***Verbmobil*** and ***Booking***.

do not always lead to the same preference operator. We detail here how, in the two corpora, some discourse markers are strong clues to recognize the binary preference operators.

The coordinating conjunction “**or**” is a strong predictor for recognizing a disjunction of preferences, at least when the “or” is clearly outside of the scope of a negation[4], as in the examples below (in π_1, the negation is part of the wh-question, and not boolean over the preference):

π_1 Why don’t we <meet, either Thursday the first>_1, or <Thursday the eighth>_2 // 1 $\triangledown$ 2

π_2 Would you like <a single>_1 or <a double>_2? // 1 $\triangledown$ 2

The preposition “**as well as**” is also a strong clue for recognizing a disjunction of preferences as in the example below.

π_3 I have <Tuesday the first>_1 open <all day>_2, as well as <the morning of May Monday the thirty first>_3 // (1 $\mapsto$ 2) $\triangledown$ 3

The coordinating conjunction “**and**” is also a strong indication, especially when it is used to link two acceptable outcomes that are both of a single type (e.g., day of the week, time of day, place, type of room, etc.) between which an agent wants to choose a single realization. For example, in ***Verbmobil***, agents want to fix a single appointment so if there is a conjunction “and” between two temporal concepts of the same level, it is a disjunction of preference (see π_4 below). It is also the case in ***Booking*** when an agent wants to book a single plane flight (see π_5).

π_4 <Monday>_1 and <Tuesday>_2 are good for me // 1 $\triangledown$ 2

π_5 You could <travel at 10am.>_1, <noon>_2 and <2pm>_3 // 1 $\triangledown$ (2 $\triangledown$ 3)

[4]When there is a propositional negation over the disjunction as in “*I don’t want Monday or Tuesday*”, we no longer have a disjunction of preferences.

The acceptability modality $\Diamond$ ($\Diamond\phi$ says that ϕ is an acceptable outcome) distributes across the conjoined NPs to deliver something like $\Diamond(\textit{meetMonday}) \wedge \Diamond(\textit{meetTuesday})$ in modal logic (clearly acceptability is an existential rather than universal modality), and as is known from studies of free choice modality (Schulz, 2007), such a conjunction translates to $\Diamond(\textit{meetMonday} \vee \textit{meetTuesday})$, which expresses our free choice disjunction of preferences, *meetMonday* $\bigtriangledown$ *meetTuesday*. We have: $(\Diamond o_1 \wedge \Diamond o_2) \leftrightarrow \Diamond(o_1 \vee o_2) \rightarrow o_1 \bigtriangledown o_2$.

On the other hand, when the conjunction "and" links two outcomes referring to a single concept that are not acceptable, it gives a conjunction of preferences, as in π_6. Once again thinking in terms of modality is helpful. The unacceptability modality $\Box$ distributes across the conjunction, this gives something like $\Box\neg o_1 \ \wedge \ \Box\neg o_2$ (where $\neg$ is truth conditional negation) which is equivalent to $\Box(\neg o_1 \ \wedge \ \neg o_2)$, i.e. $not\ o_1\ \&\ not\ o_2$ and not equivalent to $\Box(\neg o_1 \ \vee \ \neg o_2)$, i.e. $not\ o_1 \ \bigtriangledown \ not\ o_2$.

The connector "and" also involves a conjunction of preferences when it links two independent outcomes that the agent wants to satisfy simultaneously. For example, in π_7, the agent wants to book two hotel rooms, and so the outcomes are independent. In π_8, the agent expresses his preferences on two different features he wants for the hotel room he is booking.

π_6 <Thursday the thirtieth>_1, and <Wednesday the twenty ninth>_2 are, booked up // not 1 $\&$ not 2

π_7 Can I have one room< with balcony>_1 and <one without balcony>_2? // 1 $\&$ 2

π_8 <Queen>_1 and <nonsmoking>_2 // 1 $\&$ 2

Confusion between $\&$ and $\mapsto$. In this case, disagreements are mainly due to the difficulty for annotators to decide if preferences are dependent, or not. For example, in "I have a meeting <starting at three>_1, but I could meet <at one o'clock>_2", one annotator put $not\ 1 \ \mapsto \ 2$ meaning that the agent is ready to meet at one o'clock because he can not meet at three, while the other annotated $not\ 1\ \&\ 2$ meaning that the agent is ready to meet at one o'clock independently of what it will do at three.

Some connectors introduce contrast between the preferences expressed in a segment as "**but**", "**although**" and "**unless**". In the annotation, we can model it thanks to the operator $\mapsto$. When it is used between two conflicting values, it represents a correction. Thus, the annotation $o_1 \ \mapsto \ not\ o_1$ means we need to replace in our model of preferences $o_1 \succ \overline{o_1}$ by $\overline{o_1} \succ o_1$. And vice versa for $not\ o_1 \mapsto o_1$.

π_9 I have class <on Monday>_1, but, <any time, after one or two>_2 I am free. // not 1 $\mapsto$ (1 $\mapsto$ 2)

π_{10} <Friday>_1 is a little full, although there is some possibility, <before lunch>_2 // not 1 $\mapsto$ (1 $\mapsto$ 2)

π_{11} we're full <on the 22nd>_1, unless you want <a smoking room>_2 // not 1 $\mapsto$ (1 $\mapsto$ 2)

However, it is important to note that the coordinating conjunction "but" does not always introduce contrast, as in the example below, where it introduces a conjunction of preferences.

π_{12} I am busy <on Monday>_1, but <Tuesday afternoon>_2, sounds good // not 1 & 2

The subordinating conjunctions "**if**", "**because**" and "**so**" are indications for detecting conditional preferences. The preferences in the main clause depend on the preferences in the subordinate clause (if-clause, because-clause, so-clause), as in the examples below.

π_{13} so if we are going to be able to <meet that, last week in January>_1, it is going have to be <the, twenty fifth>_2 // 1 $\mapsto$ 2

π_{14} <the twenty eighth>_1 I am free, <all day>_2, if you want to go for <a Sunday meeting>_3 // 3 $\mapsto$ (2 $\mapsto$ 1)

π_{15} it is going to have to be <Wednesday the third>_1 because, I am busy <Tuesday>_2 // not 2 $\mapsto$ 1

π_{16} I have a meeting <from eleven to one>_1, so we could, <meet in the morning from nine to eleven>_2, or, <in the afternoon after one>_3 // not 1 $\mapsto$ (2 $\triangledown$ 3)

π_{17} I have seminars <all day>_1 so, maybe we should look to <next week>_2 // not 1 $\mapsto$ 2

Whether or not there are some discursive markers between two outcomes, to find the appropriate operator, we need to answer some questions: does the agent want to satisfy the two outcomes at the same time ? Are the preferences on the outcomes dependent or independent ?

We have shown in this section that it is difficult to answer the second question and there is quite some ambiguity between the operators & et $\mapsto$. This ambiguity can be explained by the fact that both operators model the same optimal preference.

Indeed, we saw in section 3.2.1 that for two outcomes o_1 and o_2 linked by a conjunction of preferences (o_1 & o_2), we have $o_1 \succ \overline{o_1}$ and $o_2 \succ \overline{o_2}$. For two outcomes o_1 and o_2 where o_2 is linked to o_1 by a conditional preference ($o_1 \mapsto o_2$), we have $o_1 \succ \overline{o_1}$ and $o_1 : o_2 \succ \overline{o_2}$. In both cases, the best possible world for the agent is the one where o_1 and o_2 are both satisfied at the same time.

3.2.3 Preference annotation scheme for non-cooperative dialogues (*Settlers* corpus)

We previously explored preference data from ***Verbmobil*** and ***Booking***, our two cooperative corpora. These corpora were relatively simple because they involved only two agents who were negotiating about how to carry out a common goal. For ***Verbmobil***, the goal was to find a meeting time, and for ***Booking*** it was to arrange a reservation. The bargaining involved particular meeting times or particular modalities of the reservation. Thus the options over which preferences were defined could be isolated by looking at noun phrases.

We extend and develop that annotation scheme to cover the more complex data provided by our non-cooperative data from the ***Settlers*** corpus. The data is more complex than that in ***Verbmobil*** or ***Booking*** because the dialogues involve typically three or more agents, each with incompatible overall goals. The need to trade requires players to form coalitions in which the participants negotiate the bargain over resources. Thus there are preferences over which coalition to form, as well as over various actions like the giving or the receiving of certain resources. We can thus not simply identify the options over which preferences are defined with the values of certain NPs in this corpus, in contrast to ***Verbmobil*** or ***Booking***.

As for ***Verbmobil*** and ***Booking***, our annotation of expressed preferences in each turn involves two steps: identify the set Ω of outcomes, on which the agent's preferences are expressed, and then identify the dependencies between the elements of Ω by using a set of specific non-boolean operators. For example, in ***Settlers***, the turn "I need clay or sheep" contains two outcomes (*the speaker's needing clay* and *the speaker's needing sheep*), when linked by a disjunction, this means the player wants one of these outcomes, preferring them equally. The same observation holds for ***Verbmobil*** (as in "Let's meet Thursday or Friday") and for ***Booking*** (as in "I would like a double room with breakfast").

In this section, we first describe how preferences can be expressed in this corpus, showing a lot of similarities compared to our first study performed on ***Verbmobil*** and ***Booking*** (see Section 3.2.1). We then show how we extend the preference annotation scheme, initially designed for these two corpora (see Section 3.2.2), to handle the more complex data. We need to extend the scheme in order to take into account preferences about different actions while in ***Verbmobil*** and ***Booking*** preferences were related to only one action corresponding to the agents' common goal (*to fix a meeting* in ***Verbmobil*** and *to book something* in ***Booking***). We then present the inter-annotator agreements on this new corpus and compare them on each corpus genre.

3.2.3.1 How preferences are expressed in the *Settlers* corpus?

We show in this section that our linguistic analysis of preferences in the ***Verbmobil*** and ***Booking*** corpus can easily be transposed to our ***Settlers*** corpus. We find again a distinction between atomic and complex preference statements with conjunctions, disjunctions and conditional dependencies.

Atomic preferences. Atomic preference statements are of the form "I prefer o_1", "Let's o_1", where o_1 describes an action. o_1 paradigmatically is identified with a verb phrase ("to trade", "to give wheat for sheep", "get an ore") or an entire clause describing an action. Sometimes o_1 is only identified by a definite noun phrase ("some of your sheep").

Preference statements can be expressed within comparatives and/or superlatives ("more clay").

Agents also express preferences using questions ("Do you want to trade?", "can i get an ore from someone?").

Preferences can also be introduced by opinions words or specific words of preferences, as in "Wood sounds good" or "I would prefer sheep". Expressions of politeness, prevalent in ***Booking***, can also be used to indirectly introduce preferences (e.g., "fresh clay if you please").

EDUs can also express preferences via modalities; "Ok now I can give you wood" tells us that it is an acceptable outcome for the agent to offer wood.

As in ***Verbmobil*** and ***Booking***, a negative preference expresses an unacceptable outcome, i.e. what the agent does not prefer. Negative preference can be expressed explicitly with negation words ("I have no wood") or inferred from the context ("I'm out too"), which means that the player rejects an offer and thus does not want to trade.

Complex preferences. Preference statements can also be complex, expressing dependencies between outcomes.

Among the possible combinations, we find again conjunctions, disjunctions and conditionals. We examine how these conjunctive, disjunctive and conditional operations over outcomes are expressed in the ***Settlers*** corpus.

Conjunctions of preferences are usually expressed via "and", as in "Can I have one sheep and one ore?", where the agent expresses two preferences (respectively over the acceptable outcomes of his getting one sheep and his getting one ore) that he wants to satisfy.

Disjunctive preference are usually expressed via "or", as in "I need sheep or wheat" or in "I can give wheat or sheep", where the agent states that either receiving (giving) sheep or (giving) receiving wheat is an acceptable outcome and he is indifferent between the choice of the outcomes.

Finally, some turns express conditional among preferences. In the ***Settlers*** corpus all offers and counteroffers express conditional preferences. In "I can wheat for sheep", there are two preferences: one for receiving sheep, and, given the preference for receiving sheep, one for the giving of wheat.

3.2.3.2 Extension of the annotation scheme

While in our ***Verbmobil*** and ***Booking*** annotation, we simply identified the outcomes with the value of certain NPs and the action in question is determined by the overall goals of meeting or booking (e.g., meeting at my office, meeting on Monday, booking an economy flight), in ***Settlers***, we need to take account of the verb to which o_1 is an argument to specify the action and the full outcome (typically verb as "trade", "give" or "receive").

In ***Verbmobil***, an outcome o_1 is typically an NP denoting a time or place; o_1 as an outcome is thus shorthand for *meet on* o_1 or *meet at* o_1. For ***Booking***, o_1 is short for *reserve* or *book* o_1.

In the ***Settlers*** corpus, preferences are more complex. An outcome o_1 can play a role in several actions: *I need* o_1 (and thus a preference for the speaker's receiving the resource o_1), *I give* o_1 (and thus a preference for offering the resource o_1), *I give* o_1 *if you give me* o_2 (and thus a preference for a trade), *I want* o_1 (and thus a preference for performing the action o_1), etc. To specify these different actions, we use, in addition to the vocabulary of our previous annotation language, two functions: *receive(o, a, <r,q>)* and *offer(o, a, <r,q>)* such that: o is the preference owner, a is the addressee, r is the resource and q is the quantity of the resource needed (or offered). If some of these arguments are underspecified, we put ?. Outcomes, which are closed under our non-boolean operators, can specify one or more arguments of our new predicates, or range over an action description.

In addition, we have decided to annotate anaphoric and unspecified bargaining moves using an empty outcome. This situation mainly occurs in case of an acknowledgement, such as "yeah me", or a rejection, such as "no I can't".

For each turn, annotators identify how outcomes are expressed and then indicate if the outcomes are acceptable, or not, using the operator *not* and how the preferences on these outcomes are linked using the operators &, $\triangledown$ and $\mapsto$.

We show below how the dialogue of example (3.3) is annotated. <*o*>_i indicates that o is the outcome number i in the EDU, the symbol // is used to separate the two annotation levels (outcome and dependency identifications) and brackets indicate how outcomes are attached.

π_1 *Euan*: And I alt tab back from the tutorial. What's up?

π_2 *Joel*: do you want <to trade>_1? // 1

π_3 *Card.*: <joel>_1 fancies <a bit of your clay>_2 // receive(1, Euan, <2,?>)

π_4 *Joel*: yes <>_1 // 1

π_5 *Joel*: !

π_6 *Euan*: Whatcha got <>_1? // 1

π_7 *Joel*: <wheat>_1? // offer(Joel, Euan, <1,?>)

π_8 *Euan*: I can <wheat>_1 for <1 clay>_2 or <1 wood>_3 // receive(Euan, Joel, <1,?>) $\mapsto$ offer(Euan, Joel, <2,1> $\triangledown$ <3,1>)

π_9 *Joel*: awesome <>_1 // 1

In annotating Joel's preference to trade with Euan in π_2, the outcome 1 describes an unspecified action (trading). In π_3, we have two outcomes that are arguments to one of our new predicates; outcome 1 (the receiver) prefers to receive from the addressee Euan the outcome 2 over any other resources. Joel accepts the trade in π_4, annotated with the anaphoric empty outcome 1. π_6 is another unspecified bargaining move where Euan's question reveals a preference for trading with Joel. Joel then proposes to Euan to give him wheat in π_7. In π_8, Euan accepts and proposes to give to Joel 1 clay or 1 wood for some wheat, annotated using the operator $\mapsto$ that links the function *receive* to the function *offer*. The free choice between outcome 2 and outcome 3 is lexicalized by the coordinating conjunction "or" and annotated with the operator $\triangledown$. As for π_4 and π_6, the annotation in the last turn π_9 provides another anaphoric preference.

3.2.3.3 Inter-annotator agreements

Two judges manually annotated three games from our corpus of 20 ***Settlers*** dialogues using the previously described annotation scheme. We performed an intermediate analysis of agreement and disagreement between the two annotators on one dialogue. The training allowed each annotator to understand the reason of some annotation choices. After this step, the dialogues of our corpora were annotated separately. Table 3.5 shows the statistics about the annotated data in the gold standard, with the dialogue used for training having been discarded.

	Settlers dialogues
No. of dialogues	74
No. of EDUs	980
No. of outcomes	632
% of EDUs with outcomes	42%
% with 1 outcome	64%
% with 2 outcomes	22%
% with 3 or more outcomes	14%
No. of not	147
No. of conjunctions (&)	20
No. of disjunctions ($\triangledown$)	27
No. of conditionals ($\mapsto$)	80

Table 3.5: Statistics for the preference annotation in ***Settlers***.

We computed four inter-annotator agreements on: (a) outcome identification, (b) outcome acceptance, (c) outcome attachment and (d) operator identification. Table 3.6 summarises our results. It also gives the agreements on the ***Verbmobil*** and ***Booking*** corpora in order to show the reliability of our extended annotation schema.

Agreements on outcome identification. We compute a *lenient* match between annotations using Cohen's kappa (see appendix A.2) (i.e. there is an overlap between their text spans as in "sheep" and "some sheep"). We obtain a kappa of 0.92.

As in ***Verbmobil*** and ***Booking***, the main case of disagreement concerns redundant preferences which we decided not to keep in the gold standard. In such cases,

	Verbmobil	***Booking***	***Settlers***
(a) (kappa)	0.85	0.85	0.92
(b) (kappa)	0.90	0.95	0.97
(c) (F-measure)	93%	82%	100%
(d) (kappa)	0.93	0.75	0.95

Table 3.6: Inter-annotator agreements for the three corpora (***Verbmobil***, ***Booking*** and ***Settlers***).

the turn does not introduce a new preference, neither does it correct the preferences stated in the previous EDUs; rather, the player just wants to insist by repeating already stated preferences, as in the following example:

π_1 Card.: anyone want wheat or wood for sheep.

π_2 Card.: wheat or wood for sheep is my seekings,

In ***Settlers***, we observed four additional cases of disagreement. The first one comes from underspecified preferences which are often used to introduce new, to make current preferences more precise or to accept preferences. Hence, we decided to annotate them in the gold standard (as shown in π_6 and π_9 in example 3.3).

The second case concerns the use of synonyms to indicate a resource (as "dolly" and "sheep"). Annotators sometimes forget to annotate a resource when it is lexicalized by a synonym.

The third case concerns disagreement on preference actions. Indeed, annotators often fail to decide if the action is about receiving or offering a resource, as in "wheat, sheep". The same lexicalizations do not always lead to the same actions. For instance, "ore for clay" may indicate, depending on the context, a request for receiving a clay or an offer to give a clay.

The last case of disagreement comes from preferences that are not directly related to the action of trading, offering or receiving a resource, as "build a settlement" in the utterance "Is there a reason I can't buy a settlement now? I have the resources". Even though those preferences were often missed by annotators, we decided to keep them, when relevant.

Agreements on outcome acceptance. The aim here is to compute the agreement on the *not* operator, that is if an outcome is acceptable, as in Dave: "I will give

<you>_1 <wheat>_2 // *offer(Dave, 1, <2,?>*", or unacceptable, as in Tomm: "No <ore>_1, sorry. *// not offer(Tomm, ?, <1,?>)*". We get a Cohen's kappa of 0.97.

As in ***Verbmobil*** and ***Booking***, the main case of disagreement concerns negations that are inferred from the context. In the example below, one annotator failed to recognize that the player Joel does not want to trade:

π_1 Card.: sorry, I'm all wheaty

π_2 Joel: same

Agreements on outcome attachment. Since the structure of the bargaining packages outcomes in a very predictable way, it is quite intuitive, and simpler than for ***Verbmobil*** and ***Booking***, to decide how options are integrated in the preference annotation in ***Settlers*** which includes functions (*offer* and *receive*). We computed annotator agreement using the F-score measure because this task involves structure building. The agreement was computed on the previously built gold standard once annotators discussed cases of outcome identification disagreements. We compared how each outcome is attached to the others within the same turn. This agreement concerns turns that contain at least three outcomes, that is 14% of the turns in ***Settlers***. For example, in "Joel wants to trade wheat for clay, or wheat for ore" the annotation should be something like: (*receive(Joel,?,<clay,?>)* $\mapsto$ *offer(Joel,?,<wheat,?>)*) $\bigtriangledown$ (*receive(Joel,?,<ore,?>)* $\mapsto$ *offer(Joel,?,<wheat, ?>)*) where brackets indicate how outcomes are attached. We obtained a perfect agreement.

Agreements on outcome dependencies. Finally, we computed the agreements for each couple of outcomes on which annotators agreed about how they are attached. In our ***Settlers*** corpus, the most frequent operators are not and $\mapsto$ because the main purpose of the players in this corpus is to propose, accept or reject a trade. The other two operators $\&$ and $\bigtriangledown$ are equally split. The most frequently used binary operators were $\mapsto$ in ***Verbmobil*** and & and $\mapsto$ in ***Booking***.

Table 3.7 shows the kappa for each operator on each corpus genre. The Cohen's kappa, averaged over all the operators, is 0.93 for ***Verbmobil***, 0.75 for ***Verbmobil*** and 0.95 for ***Settlers***. In ***Verbmobil*** and ***Booking***, we observed two main cases of disagreement: between $\bigtriangledown$ and &, and between & and $\mapsto$. These cases were more frequent for ***Booking***, accounting for the lower kappa there than for ***Verbmobil***. In ***Settlers***, the main case of disagreement concerns the confusion between $\bigtriangledown$ and &. The high agreement on $\mapsto$ reflects the fact that $\mapsto$ occurs in the description of an offer which is easy to annotators to spot.

	Verbmobil	***Booking***	***Settlers***
&	0.90	0.66	0.88
$\bigtriangledown$	0.97	0.89	0.93
$\mapsto$	0.92	0.71	1.00

Table 3.7: Agreements on binary operators for the three corpora (***Verbmobil***, ***Booking*** and ***Settlers***).

As we said in Section 3.2.2.2, the confusion between $\bigtriangledown$ and & is mainly due to the same linguistic realizations that do not always lead to the same annotations. For instance, for π_1 below, we have two different annotations:

π_1 *Dave*: willing to trade for <clay>_1 and <sheep>_2

* Annotation 1: *receive(Dave, ?, <1,?> & <2, ?>)*
* Annotation 2: *receive(Dave, ?, <1,?>* $\bigtriangledown$ *<2, ?>)*

In ***Settlers***, as it is also the case in ***Verbmobil*** and ***Booking***, the coordinating conjunction "**or**" is a strong predictor for recognizing a disjunction of preferences, at least when the "or" is clearly outside of the scope of a negation.

In ***Verbmobil*** and ***Booking***, the coordinating conjunction "**and**" is also a strong indication, especially when it is used to link two acceptable outcomes that are both of a single type (e.g., day of the week, type of room, type of resource, etc.) between which an agent wants to choose a single realization. For example, in ***Verbmobil***, agents want to fix a single appointment so if there is a conjunction "and" between two temporal concepts of the same level, it is a disjunction of preferences. It is also the case in ***Booking*** when an agent wants to book a single plane flight.

In ***Settlers***, the connector "and" generally links two outcomes that the agent wants to satisfy simultaneously and involves a conjunction of preferences, as in Dave: "I can give <you>_1 <one wheat>_2 and <ore>_3 for <wood>_4" where we have: *receive(Dave, 1, <4, ?>)* $\mapsto$ *offer(Dave, 1, <2, 1> & <3, ?>).*

When the conjunction "and" links two outcomes and one at least is unacceptable, it gives a conjunction of preferences, as in Dave: "I dont have <any ore>_1, but i do have <plenty clay>_2" where we have: *not offer(Dave, ?, <1, ?>) & offer(Dave, ?, <2, ?>).*

3.3 Conclusion

In this chapter, we studied how preferences are linguistically expressed in elementary discourse units on different corpus genres. We first investigated preferences within negotiation dialogues with a common goal like fixing a meeting time (in our ***Verbmobil*** corpus) or making a hotel or plane reservation (in our ***Booking*** corpus). We then studied preferences in the complex domain of ***Settlers***, where the types of actions were more diverse.

Our preference annotation scheme required two steps: (1) identify the set of acceptable and non acceptable outcomes on which the agent's preferences are expressed, and then (2) identify the dependencies between these outcomes by using a set of specific non-boolean operators expressing conjunctions, disjunctions and conditionals. While in ***Verbmobil*** and ***Booking***, we simply identified the outcomes with the value of certain noun phrases, in ***Settlers*** we used, in addition to the vocabulary of our previous annotation language, two functions (receive and offer) to handle more complex preferences.

The inter-annotator agreement study shows good results on each corpus genre for outcome identification, outcome acceptance and outcome attachment. For outcome dependencies, annotators had to face some difficulties especially in the ***Booking*** corpus. Difficulties concern the confusion between disjunctions and conjunctions mainly because the same linguistic realizations do not always lead to the same operator. In addition, annotators often fail to decide if the preferences on the outcomes are dependent or independent.

This work shows that preference acquisition from linguistic actions is feasible for humans and that our scheme adapts relatively easily to different domains. In the next chapter, we present how to automate the process of preference extraction from elementary discourse units using Natural Language Processing methods.

Chapter 4

Preference extraction

4.1 Outcome extraction **110**

4.1.1 External resources 110

4.1.2 Classifier and feature set 112

4.1.3 Experiments and results 114

4.2 Preference identification **116**

4.2.1 Outcome acceptance 117

4.2.2 Outcome attachment 118

4.2.3 Outcome dependencies 119

4.3 Conclusion **119**

According to the linguistic study of preferences from the previous chapter, we present an NLP-based approach to extract preferences. Similarly to the annotation process, we perform two steps: (1) we extract the set of outcomes using machine learning techniques with a combination of local and discursive features (see Section 4.1), (2) we then identify the preferences over the outcomes by using an hybrid approach combining both machine learning techniques (for outcome acceptance) and rule-based approaches (for outcome attachment and outcome dependencies) (see Section 4.2). For each subtask, we assess the reliability of our method on both ***Verbmobil*** and ***Booking*** corpora.

4.1 Outcome extraction

The problem of outcome extraction is to decide whether a given token is an outcome or not. Hence, the issue is to classify tokens into two categories: "*Outcome*" and "*Non-Outcome*". We recall that outcomes can be noun phrases, prepositional phrases or verbal phrases. We thus need to choose which token or group of tokens have to be classified.

In the data, agents negotiate to reach an agreement on an action: to meet on a specified day, to book a certain flight, etc. We are generally informed about these actions in verbal phrases. However, terms corresponding to preference outcomes are rather contained in Noun Phrases (NP). For example, to schedule an appointment, negotiation deals with days and times. To book a hotel or a plane ride, negotiation deals with more specific options such as "a direct flight", "a double room". Therefore, it is appropriate to extract noun phrases (with prepositions that precede as they are unavoidable when discussion focuses on concepts of place or time).

To extract NPs, we use the Charniak's syntactic parser (Charniak, 2000). We look down through the syntactic tree of each EDU in order to find the smallest NP, preceded by a preposition if applicable. If a PP node subsumes this NP, we extract the PP node instead. For example, in the following syntactic tree, *(PP (IN in) (NP (DT the) (NN afternoon)))*, we extract "*in the afternoon*". Using this method, the following NPs and PPs are extracted from the ***Verbmobil*** example presented below: for π_3, "*I*", "*some free time*", "*almost every day*", "*except Fridays*"; for π_4, "*Fridays*"; for π_5, "*In fact*", "*I*" and "*on Thursday*".

(4.1) π_3 B: I have some free time on almost every day except Fridays.
π_4 B: Fridays are bad.
π_5 B: In fact, I'm busy on Thursday too.

Before presenting our machine learning approach to outcome extraction, we first describe the external resources that we use.

4.1.1 External resources

Our approach needs two kinds of resources: a domain ontology and a lexicon.

4.1.1.1 Domain Ontology

Our data are from two different domains: the domain of temporal designations for ***Verbmobil*** where agents discuss about a date (*days*, *months*) and a time (*afternoon*, *morning*, *at 2 p.m.*, etc.) to schedule a meeting, and the tourism domain for ***Booking*** where agents have to book flights, rooms etc. The tourism domain also deals with temporal concepts, since agents have to choose dates of booking. Ideally, we need two domain ontologies. Since our goal is to study the impact of discourse structure on outcome extraction and to what extend our method is domain-dependent, we decided to use one domain ontology. We therefore designed our features (see the next section) to handle outcomes exclusively for ***Verbmobil*** and then to evaluate how our approach performs on a new corpus genre (see section 4.1.3).

Our domain ontology has to model a calendar (time, days, etc.). To create it, we used upper (or top-level) ontologies that describe very general concepts that are domain independent. Two top-level ontologies were used: SUMO (Suggested Upper Merged Ontology)[1] and COSMO (COmmon Semantic MOdel)[2]. We grouped the relevant concepts of both ontologies, then we changed the hierarchy to better fit our problem. To this end, we removed a number of classes that are not useful for our work (e.g. *Sunrise*, *Sunset* or *ColdSeason*) and we enhanced the ontology with other useful concepts (e.g. *Evening* which completes *Morning* and *Afternoon*). We also added properties and axioms (for example, a disjunction to say that an instance of *Monday* cannot be an instance of *Tuesday*, *Wednesday*, or *Sunday*). These axioms do not help in outcome extraction but they are useful for reasoning about preferences that follows (see Section 5.2.2). Our domain ontology has been implemented under the ontological engineering tool Protégé[3] and actually contains 43 concepts with one label for each, 26 object properties and 7 data properties.

4.1.1.2 Lexicon

In addition to the domain ontology, our classifier needs lexical knowledge. As described in section 3.2.1, preferences often co-occur with modals (*Tuesday would be good*), negations (*I am not free on Monday*) and opinion words that indicate whether the agents' preferences are acceptable or not (*good*, *bad*, *OK*, *like*, etc.). Preferences can also be introduced by polite words (*please*, *thanks*, etc.) as well as by specific words of preferences, such as verbs (*to prefer, to favour*, etc.), adjectives (*favourite, preferable*, etc.), nouns (*choice, predilections, druthers*, etc.) and adverbs (*too, rather*, etc.).

To take into account these clues, we created a domain-independent lexicon that contains 10 modals, 14 negations, 5 polite words, 170 opinion words (20 verbs, 54 adjectives, 22 nouns and 74 adverbs) and 31 preference words (9 verbs, 6 adjectives,

[1] http://stuarthendren.net/resources/sumodlfull.owl
[2] http://micra.com/COSMO/COSMO.owl
[3] http://protege.stanford.edu/

13 nouns and 3 adverbs). For opinion words, negations and modals we reuse an already existing lexicon (Benamara et al., 2011).

4.1.2 Classifier and feature set

To classify each NPs into the classes "*Outcome*" or "*Non-Outcome*", we used two categories of features: local features and discursive features. All the features are binary. The classifier is based on SVMs ("Support Vector Machines") (Burges, 1998). A feature vector is computed for each NP within an EDU.

4.1.2.1 Local features

The scope of these features is either the unit to be classified, namely an NP, or the segment that contains this NP.

We have five features at the NP level that test if the NP contains: (1) a lexicalization of a concept that belongs to the domain ontology, (2) a comparative, (3) a superlative, (4) a disjunction or (5) a conjunction.

We have ten features at the segment level:

- (1) *The left context of the NP is a lexicalization of a concept that belongs to our domain ontology.* Since the list of terms associated to each concept in our ontology is small, this feature helps us to detect additional lexicalizations (as for the NP *the twenty seventh* in the segment *the best time for me would be next Monday the twenty seventh*).

- (2-3) *The segment contains a disjunction or a conjunction.* Sometimes, conjunctions and disjunctions are not part of the NP to be classified, hence, the two features that look for them at the NP level will not suffice (as for *the eleventh* and *the twelfth* in *(NP (DT the) (JJ eleventh)) (CC or) (NP (DT the) (JJ twelfth)))*).

- (4-5) *Scoping features.* We have three features. The first two ones look if the NP is under the scope of a negation or a modal. Scope is resolved quite simply using the syntactic tree of an EDU. We consider that an NP is in the scope of a negation or a modal word if the father node of that word is also a father of the NP node. Of course, this procedure does not suffice for resolving some scoping ambiguities, especially for negation, as in *excepting this week because I am on vacation* where the scope of the negation is the entire sentence. However, since segments in our data are quite short, this simple approach seems to give correct results in most of the cases.

(6) As agents negotiate to reach an agreement about a specific action (to meet, to make a reservation), the third feature looks if the NP is in the scope of a domain action verb. To do that, we use a closed list of verb ("*to meet*" in ***Verbmobil***, "*to book*" and "*to reserve*" in ***Booking***).

- (7-9) *The segment contains an opinion word, polite words or words that introduce preferences*, as encoded in our lexicon (see section 4.1.1).
- (10) *The segment contains a preference of another agent.* The last feature indicates if there is a reference to the other agent in the segment, as in: *you say you do not have anything open, Thursday morning, or Wednesday afternoon?*, where the agent does not bring new information about preferences but only repeats the already stated preferences of the other agent. This feature accounts for the "*Non-Outcome*" class.

4.1.2.2 Discursive features

We have nine discursive features.

- (1-6) *Features that use the rhetorical relations that link the current EDU to the segments that precede or follow it.* We noticed that some discourse relations can help highlight segments that contain, or not, preferences. For example, in a segment introduced by an *Elaboration*, there are chances of finding preferences. On the contrary, in a *Comment* segment, NPs most likely belong to the "*Non-Outcome*" class. Thus, we split discourse relations into three categories: (a) those that "generally" imply a "*Non-Outcome*" (*Explanation*, *Comment*, *Clarification Question*, *Summary* and *Acknowledgment*), (b) those that *may* involve a "*Outcome*" (*Elaboration*, *Continuation*, *Indirect QAP*, *Correction*, *Contrast*, *Alternative*, *Consequence*, *Result* and *Narration*), and (c) those that "generally" involve a "*Outcome*" (the list of relations that we used to annotate the corpora does not include such a relation). In ***Verbmobil***, 86% of discourse relations are in category (a), while 14% are in category (b). We observe the same trend for ***Booking***.

  We thus have six features: three that test whether the relation that links the current EDU to the previous one belongs to one of our three categories (non-outcome, outcome and possible outcome) or not. The other three concerns the relation between the current EDU and the next one.
- (7-8) *The current EDU or the segment that precedes it is a question.* Interrogative forms are often used during the negotiation in order to introduce agents' preferences in a roundabout way and/or to query the other agent on its own

preferences. In our corpus, interrogatives are not always followed by a question mark. To detect questions, specific rhetorical relations are used, such as *QAP*, *Q-Elab*, *Continuation Question*, *Clarification Question*.

- (9) *Frequency.* The last feature tests if the NP occurs at least twice in the dialogue.

4.1.3 Experiments and results

Several experiments were performed for testing the validity of our extraction approach. The first experiment was carried out on our ***Verbmobil*** corpus. The training corpus consisted of 25 dialogues, i.e. 2374 NPs, and the test corpus consists of 10 dialogues, i.e. 700 NPs. In the second experiment, the classifier was trained on 15 dialogues from our ***Booking*** corpus i.e. 837 NPs and tested on 6 dialogues with a total of 312 NPs. Finally, the classifier was evaluated using ***Verbmobil*** for training (using the 35 dialogues) and ***Booking*** for test (using the 21 dialogues) (`Vb + Bk`). The latter, rather unusual, test configuration is supposed to help determine whether our method allows for training on a larger, already available annotated corpus and testing on smaller one, sometimes from a different domain. For all setups, we used the SVM-light software package[4].

We compared the results of the classifier with those of three baselines: (1) the first one classifies all the NPs in the "*Outcome*" category, (2) the second one classifies in the "*Outcome*" class all the NPs that contain a concept belonging to the ontology, finally (3) the third baseline is a simplified version of our classifier that only uses a subset of our features (we removed features based on ontology as well as all the features that are based on discourse relations).

Table 4.1 shows the precision, recall and F-measure for each configuration. It first presents the results of the baselines. We then develop our model first by considering NP local features, then adding local features for the segment and then progressively adding discourse features (the addition is marked by the "+" sign). The last row presents the final results, obtained by using all features.

The results in Table 4.1 show that, among the three baselines, the second one provides the best results for ***Verbmobil***. This is expected, since the ontology is tuned to these data. However, it has limitations, because some NPs that contain a concept of the ontology are not outcomes (since they are repetitions, comments, etc.) and of course not all the outcomes expressed by agents are "covered" by concepts in the ontology. For ***Booking***, the ontology degrades the results (namely, the recall) with respect to the first baseline, since there is a weak overlapping between

[4] `http://svmlight.joachims.org/`

		Verbmobil			*Booking*			Vb + Bk		
		P	R	F	P	R	F	P	R	F
Baselines	All the NP	40.9	100.0	58.1	28.0	100.0	43.8	28.3	100.0	44.1
	Ontology alone	95.6	61.3	74.7	55.6	16.7	25.7	49.2	13.5	21.2
	Simple classifier	65.2	71.1	68.0	68.4	43.3	53.1	43.9	55.7	49.1
Local Features	All features (NP)	95.7	62.0	75.2	100.0	3.3	6.5	50.7	16.0	24.4
	+ All features (Segment)	94.1	78.9	85.8	68.4	43.3	53.1	60.2	26.2	36.5
Discursive Features	+ Previous Relation	94.9	78.9	86.2	67.6	41.7	51.6	60.2	26.2	36.5
	+ Following Relation	94.0	77.5	84.9	66.7	40.0	50.0	59.4	25.3	35.5
	+ Questions	95.6	75.4	84.3	79.0	50.0	61.2	59.4	25.3	35.5
	+ $\geq$ 2 occurrences of the NP	90.8	83.1	**86.8**	75.6	56.7	**64.8**	62.9	32.9	**43.2**

Table 4.1: Results for automatic preference extraction.
P, R and F are the Precision, Recall and F-measure.

the concepts in the ontology and those in this corpus. The same goes for the third test (Vb + Bk). However, this is not a critical issue in principle, since suitable ontologies are available for the ***Booking*** domain as well. In all cases, the third baseline provides quite stable results, consistently better than the first baseline and, in the second and third tests (for which no suited ontology was used) better than the second baseline as well. Interestingly, the simple classifier yields a better recall for the third test than for the second one. This might point out a data sparsity problem in training on ***Booking*** only (the ***Booking*** configuration).

The results show quite similar behaviours of the method on both ***Verbmobil*** and ***Booking***. We see that the local features at the NP level are relevant for obtaining a good precision. This is especially well-marked in ***Verbmobil***. The segment-level and the discursive features improve the recall and F-measure in all three test configurations. The improvement is more marked in the second and third tests, where the precision is increased as well. This might be because the ontology, less suited to these tests, has a lower impact on the performance figures. Finally, for ***Verbmobil***, we obtain an F-measure of 86.8 %, i.e. almost 20 % above the third baseline (simple classifier) and more than 10 % above the second baseline (based on the ontology). For ***Booking***, we obtain an F-measure of 64.8 %, i.e. more than 10 % above the simple classifier baseline. For the third test, the results do not show improvement over baselines. This is probably caused by the influence of the ontology, which better fits the support vectors to the training corpus (***Verbmobil***), making them less relevant to the test corpus. When we disable the two ontology-based features, we obtain a precision of 50.2 %, a recall of 62.9 % and an F-measure of 55.8 %, hence, an improvement over the baselines.

As for the discursive features, we notice that, for ***Verbmobil***, the rhetorical relation between the current EDU and the previous one yields a more important improvement than other discourse information. This could be explained by the nature of the corpus, where task context (as expressed in previous dialogue turns) is important. For ***Booking***, the current EDU or the segment that precedes being question yields the most salient performance improvement. This could also be explained by the nature of the corpus, which mainly contains question-answer pairs at a dialogue level. For the third test, discursive features do not bring a consistent improvement over the baselines. This is perhaps caused by the inability of discourse information to compensate for the mismatch between training and test data: indeed, in principle there are more instances of local features (at the NP and EDU level) associated to positive examples, than of discursive features associated to positive examples; and when the classifier is trained on features extracted from a corpus domain and tested on another corpus domain, the weight of the discursive features might not suffice to compensate for the other, local, features. In all three test configurations, the feature testing for the presence of an NP at least two times in a dialogue yields consistent improvements over all other features. This is somehow expected, since, in principle, NP frequency provides topicality information, and it makes sense that preferences tend to be expressed on the main topic of a discourse.

While the problem of extracting discourse structure remains formidable, we can approximate these relations relatively well for our purposes using features that can be conveniently obtained automatically, e.g. the presence of questions. Others like the type of discourse relations relating the current EDU to prior segments and to the EDU to come depend on an automated ability to recognize discourse relations. This is not the hardest task in discourse parsing and the prognosis is relatively optimistic (Baldridge and Lascarides, 2005b; Wellner et al., 2006). Our study here shows the importance of discourse features for preference extraction, assuming that these are given by manual annotation.

4.2 Preference identification

Once we extracted the set O of outcomes from each EDU, the next step was to identify how these outcomes are ordered. To achieve this goal, we performed three subtasks: (1) first, we identify the set of unacceptable outcomes. This comes down to associate, or not, the operator *not* to each element in O. For example, from π_1: "I have got a class <on Tuesday>_1 and <Thursday>_2 <from nine to twelve>_3", we get: 1, 2, not 3; (2) the next step is, for each EDU that contains more than one outcome (around 45% of the EDUs that contain outcomes), to provide a structured representation of elements in O in order to get elementary couples of outcomes. This leads to the following representation for π_1: $((1,\ 2),\ not\ 3)$; (3) finally, for each couple of outcomes, we recursively identify the operator that links them. For instance, for π_1 we get: $((1,\ \triangledown,\ 2),\ \mapsto,\ not\ 3)$.

Tables 4.2 and 4.3 present the results for the evaluation of the subtasks.

	Verbmobil	*Booking*
Outcome acceptance	89%	
Outcome attachment	81%	75%
Operator identification	83%	59%

Table 4.2: Results for automatic preference identification.

	Verbmobil	*Booking*
&	88%	38%
$\triangledown$	96%	71%
$\mapsto$	96%	69%

Table 4.3: Results for outcome dependencies.

4.2.1 Outcome acceptance

In order to decide whether an outcome is acceptable or not, we performed a binary classification task. Unacceptable outcomes are generally in the scope of word negators (*no, not*), negative opinion words (*bad*), some expressions (*I have meetings, I got classes*) or inferred from the context without any lexicalization.

Inspired from recent efforts in this field (Jia et al., 2009; Li et al., 2010), we designed a set of nine features: (1) the EDU contains a negation, (2) the outcome is in the scope of the negation, (3) there is a delimiter between the negation word and the outcome in order to eliminate some words from the scope, (4) the number of negation words, (5) the number of outcomes in the EDU, (6) the syntactic categories of the term associated to the outcome and (7) of the negation word, (8) the label of the negation word and finally (9) the number of tokens between the object being classified and the negation word.

We carried out a 10-fold cross-validation on both ***Verbmobil*** and ***Booking*** using a Maximun Entropy algorithm[5]. We get an F-measure of 89%.

Errors concern scoping errors due to parsing and implicit negations, as in "<Tuesday>_1 I have got a meeting <from one to three>_2 and then another one <from four to six>_3" where 3 is classified as an acceptable outcome.

4.2.2 Outcome attachment

To perform the subtask of outcome attachment, we rely on a symbolic approach. We note that, within the structured representation, outcomes are ordered according to how their corresponding nodes are linked in the syntactic tree. In π_1, the NPs "on Tuesday" and "Thursday" are objects of the verb "got" and thus have the same direct father node whereas the NP "from nine to twelve" belongs to another node within the tree. We have then the couples $((1,\ 2),\ 3)$.

However, in some cases, this order has to be reversed mainly for two reasons: (1) the presence of specific discourses cues, such as "if" and "because", as in "<the twenty eighth>_1 I am free, <all day>_2, if you want to go for <a Sunday meeting>_3", where we have $(3,\ (1,\ 2))$ since the annotation should be $3 \mapsto (1 \mapsto 2)$; (2) the outcomes are not at the same ontological level, such as a day and a period of time, as in "yeah <the afternoon>_1 is okay, <on Wednesday>_2" where we have $2 \mapsto 1$.

We also note that in case of some discourse cues that introduce a contrast (as "but", "although"), the syntactic order has to be modified, as in "I have class <on Monday>_1, but, <any time, after one or two>_2 I am free" where we have $(1,\ (1,\ 2))$, since the annotation should be $not\ 1 \mapsto (1 \mapsto 2)$. Detecting contrasts is not easy, particularly when markers are ambiguous as "but" which sometimes involves contrast (see previous example) and sometimes not as in "I have a meeting, <starting at three>_1, but I could meet <at, one o'clock>_2" where we have $not\ 1\ \&\ 2$.

The rules were built according to the same development set as for outcome extraction, i.e 25 ***Verbmobil*** dialogues, and were evaluated on a test set of 31 dialogues (10 from ***Verbmobil*** and 21 from ***Booking***) that contains 412 elementary outcome couples. The F-measure is 81% for ***Verbmobil*** and 75% for ***Booking***.

These results are in good agreement with the results we obtained on outcome attachment during the annotation (see Section 3.2.2.2). Errors come both from the parser (especially for coordination attachment) and from the difficulty of detecting contrasts.

[5] http://nlp.stanford.edu/software/classifier.shtml

4.2.3 Outcome dependencies

The last step in our process was to identify how the two outcomes from a couple are related using the operators $\bigtriangledown$, $\&$ and $\mapsto$.

As for outcome attachment, we performed this step using a set of rules designed exclusively by using 25 dialogues from ***Verbmobil*** and then assessed on 31 dialogues from both ***Verbmobil*** and ***Booking***.

We got the following F-measures (results are given in the form (***Verbmobil***, ***Booking***): (88%, 38%) for $\&$, (96%, 71%) for $\bigtriangledown$, and (96%, 69%) for $\mapsto$ which correspond to an average score of 93% for ***Verbmobil*** and 59% for ***Booking***.

As for humans, our system sometimes fails to distinguish between $\&$ and $\mapsto$, between $\bigtriangledown$ and $\mapsto$, and between $\&$ and $\bigtriangledown$. Errors are more frequent for ***Booking*** because of the nature of the dialogues (more than one sentence per segment, compared to ***Verbmobil*** where segments are smaller). This makes the identification of dependencies among outcomes from distinct sentences more difficult. The errors in ***Booking*** are also due to a less clear correspondence between linguistic cues and our operators (see the discussion at the end of Section 3.2.2.2).

Given that the preferences $o_1 \mapsto o_2$ and $o_1 \,\&\, o_2$ yield the same set of best outcomes, the agent is ready to act so that o_1 and o_2 are both realized. We have thus decided to collapse these two operators in order to extract, from each EDU, the preference for the best outcome. This leads to higher average F-measures of 98% for ***Verbmobil*** and 81% for ***Booking***.

4.3 Conclusion

In this chapter, we presented an NLP-based approach to preference extraction from negotiation dialogues. We proposed to extract preferences in two steps. First, we used a machine learning approach that extracts outcome expressions from dialogues using a combination of local and discursive features. Then we used a hybrid approach in order to identify the preferences over the outcomes.

We assessed the reliability of our method on the ***Verbmobil*** and ***Booking*** corpora. For outcome extraction, our results showed that the dialogue discourse structure coupled with a top-level ontology are helpful to efficiently extract preferences. Our study here showed the importance of discourse features assuming that these are given by manual annotation. In future work, we plan to recognize discourse structure automatically. For preference identification, for each subtask, the results are in good agreement with the results obtained for the manual annotation (see Chapter 3).

Now we have seen how to automatically extract preferences, we want to give a formal description of each agent's preferences. In the next chapter, we propose a procedure to translate the preference operators into CP-nets. This provides each operator with a well-defined semantics. We also provide a method showing how CP-nets from dialogue segments combine via discourse structure to provide a model of agent preferences at any moment in the dialogue. Our model also shows the evolution of these preferences as the dialogue progresses.

Chapter 5

Discourse and preference evolution

5.1 CP-net descriptions **123**

5.2 Rules to construct preferences from EDUs **124**

5.2.1 From preferences in EDU together with rhetorical relations to partial preference descriptions 124

5.2.2 From preference descriptions to models 134

5.2.3 Treatment of an example 135

5.3 Evaluation of the proposed method **139**

5.4 Conclusion **142**

Dialogues are structured by various moves that the participants make—e.g., answering questions, asking follow-up questions, elaborating prior claims, and so on. Such moves often affect the way interlocutors view a speaker's preferences and consequently influence how they respond. So mapping dialogue moves to *preferences* is an important task: for instance, they are vital in decisions on how to re-plan and repair should the agents' current plan fail, for they inform the agents about the relative importance of their various goals. Classical game theory, however, demands a complete and cardinal representation of preferences for the optimal intention to be defined (see Section 1.2). This is not realistic for modelling dialogue because agents often lack complete information about preferences prior to talking: they learn about the domain, each other's preferences and even their own preferences through dialogue exchange. For instance, utterance (5.1) implies that the speaker wants to go to the mall given that he wants to eat, but we do not know his preferences over "go to the mall" if he does not want to eat.

(5.1) I want to go to the mall to eat something.

What's required, then, is a method for extracting partial information about preferences and the dependencies among them that are expressed in dialogue, perhaps indirectly, and a method for exploiting that partial information to identify the next optimal action. This chapter proposes a method that builds on (Asher et al., 2010) for achieving these tasks by exploiting discourse structure.

We studied 20 dialogues chosen at random from our ***Verbmobil*** corpus and annotated according to the Segmented Discourse Representation Theory, SDRT (see Section 3.1.1). Across the corpus, more than 30% of the discourse units are either questions or assertions that help elaborate a plan to achieve the preferences revealed by a prior part of the dialogue—these are marked respectively with the discourse relations *Q-Elab* and *Plan-Elab* in SDRT, and utterances π_2, π_6 and π_7 and the segment π_3-π_6 in the ***Verbmobil*** example 3.1 invoke these relations (for clarity, we recall the ***Verbmobil*** example below). Moreover, 10% of the moves revise or correct prior preferences (like π_5), 6% of them explain prior content or prior moves (like π_4) and more than 20% of the discourse units are marked with *Continuation* that means they reflect effects of the continued relation. The remaining 35% either do nothing or have the same effect on preferences as *Elaboration*.

(5.2) π_1 A: Shall we meet sometime in the next week?

π_2 A: What days are good for you?

π_3 B: I have some free time on almost every day except Fridays.

π_4 B: Fridays are bad.

π_5 B: In fact, I'm busy on Thursday too.

π_6 A: Next week I am out of town Tuesday, Wednesday and Thursday.

π_7 A: So perhaps Monday?

We will model the interaction between dialogue content and preferences in two steps. The first maps utterances and their rhetorical connections into a partial description of the agents' preferences. The mapping is *compositional* and *monotonic* over the dialogue's logical form (i.e., the description of preferences for an extended segment is defined in terms of and always subsumes those for its subsegments): it exploits recursion over discourse structure. As explained in Section 1.3, we adopt CP-nets to model preferences because they provide a compact, computationally efficient, qualitative and relational representation of preferences and their dependencies, making them compatible with the kind of partial information about preferences that utterances reveal. They also provide a computationally highly efficient method for identifying the optimal intention from the various (perhaps conflicting) preferences via the linear *forward sweep algorithm*. Our mapping from the logical form of dialogue to partial descriptions of Boolean CP-nets proceeds in a purely linguistic or domain independent way (e.g., it ignores information such as Monday and Tuesday cannot co-refer) and will therefore apply to dialogue generally and not just ***Verbmobil***.

In a second stage, we “compress” and refine our description making use of constraints proper to CP-nets (e.g., that preference is transitive) and constraints provided by the domain—in this case constraints about times and places, as well as constraints from deep semantics. This second step reduces the complexity of inferring which CP-net(s) satisfy the partial description and allows us to identify the minimal CP-net that satisfies the domain-dependent description of preferences. We can thus exploit dependencies between dialogue moves and mental states in a compact, efficient and intuitive way.

We start by motivating and describing the semantic representation of dialogue with CP-net descriptions in Section 5.1 and then describe the rules to construct CP-nets from Elementary Discourse Units (EDUs) in Section 5.2. We then test the algorithms predictions against the judgements of naive annotators on three random unseen dialogues. Section 5.3 reports on these results.

5.1 CP-net descriptions

Dialogue turns sometimes inform us that certain variables enter into preference statements. We'll express the fact that the variables $x_1, \ldots, x_n$ are associated with discourse constituent π by the formula $x_1, \ldots, x_n(P(\pi))$, where $P(\pi)$ refers to the partial description of the preferences expressed by the discourse unit π (see Section 5.2).

Although CP-nets generally consider variables with a finite range of values, to define the mapping from dialogue turns to descriptions of CP-nets in a domain independent and compositional way, we use *Boolean* propositional variables: each variable describes an action that an agent can choose to perform, or not. We will then refine the CP-net description by using domain-specific information, transforming CP-nets with binary valued variables to CP-nets with multiple valued variables (see Section 5.2.2). This reduces the complexity of the evaluation of the CP-net by a large factor.

As presented in Section 1.3.3, the conditional preference tables (CPTs) in CP-nets associate a total order with each instantiation of parents. To deal with partial preferences, we propose a description language for CP-nets in which not every portion of the the CPTs needs to be specified; consequently, the formulas in this description denote a set of CP-nets or a partial CP-net. For instance, the description language formula $w \succ y(CPT(X))$ describes any complete CP-net $\mathcal{N}$ where $CPT(X) \in T_{\mathcal{N}}$ contains an entry of the form $p : w \succ y$ for some possibly empty list of parent variables p. And $x_1, \ldots x_n : w \succ y(CPT)$ describes any complete CP-net $\mathcal{N}$ where some CPT in $T_{\mathcal{N}}$ contains the entry $p : w \succ y$, where $x_1, \ldots, x_n$ are conjuncts in p. A CP-net description $\mathcal{DN}$ is a set of such formulas (and so the complete CP-net $\mathcal{N} \models \mathcal{DN}$ just in case $\mathcal{N}$ satisfies each description language formula in $\mathcal{DN}$).

The preference information captured by a CP-net description $\mathcal{DN}$ can be viewed as a set of logical assertions about a user's preference ordering over complete assignments to variables in the network. These statements are generally not complete, that is, they do not determine a unique preference ordering. Thus a partial CP-net $\mathcal{DN}$ defines a partial preference order over outcomes: neither $o \succeq o'$ nor $o' \succeq o$ may follow from $\mathcal{DN}$, making o and o' incomparable until $\mathcal{DN}$ is refined into a more specific description.

The description language allows us to impose constraints on the CP-nets that agents commit to without specifying the CP-net completely, as is required for utterances like (5.1).

5.2 Rules to construct preferences from EDUs

In Section 5.2.1, we describe rules to construct the CP-net description from $P(\pi)$, the partial description of the preferences expressed by the elementary discourse unit (EDU) π, and from the rhetorical relations linking the EDUs.

In Section 5.2.2, we describe how to construct a minimal CP-net from a satisfiable CP-net description. One can then use the *forward sweep* procedure for outcome optimization.

We apply all these rules on a complete example in section 5.2.3.

5.2.1 From preferences in EDU together with rhetorical relations to partial preference descriptions

During the dialogue, the knowledge about each agent's preferences change according to what they say. In order to formalize this evolution, we construct a partial description of CP-nets $\mathcal{DN}$ from the discursive structure of the dialogue. The rhetorical relations between EDUs allow to compositionally construct this preference description by imposing different constraints, reflecting the fact that the semantic relations between segments influence relations between preferences. We present here the rules for each rhetorical relation used in the annotation of the ***Verbmobil*** corpus. We will add rules for *Commit(*π*,* $\mathcal{DN}$*)* to define the agent's *Commitment* on the content of the discourse units π which introduce preferences.

In these rules, X, Y and Z denote propositional variables on which preferences are expressed and $x, \overline{x}, y, \overline{y}, z, \overline{z}$ are their associated values. ϕ and ψ denote propositional formulas standing for complex preferences and $Var(\phi)$ is the list of variables in ϕ. $Pa(X)$ is the set of parent variables of X and $\succ_X$ is the preference relation which describes the conditional preference table $CPT(X)$.

These rules deal with all the kind of complex preferences encountered during preference annotation (see Section 3.2). We recall the semantic of the preference operators. *not* is used to represent negative preference, thus $not\ o_1$ means that o_1 is an unacceptable outcome. $\&$, $\bigtriangledown$ and $\mapsto$ represent respectively conjunctions, disjunctions and conditionals. $o_1\ \&\ o_2$ means that the agent wants to satisfy both o_1 and o_2 and he prefers to have one of them if he can not have both. $o_1\ \bigtriangledown\ o_2$ means that the agent wants to satisfy o_1 or o_2 and he is indifferent as to the choice of which one. $o_1\ \mapsto\ o_2$ means that the agent wants to satisfy o_1 and if o_1 is satisfied he wants to satisfy o_2 too. An other preference operator is introduced in these rules, Δ which represents a strong conjunction. $o_1\ \Delta\ o_2$ means that the agent prefers to satisfy both o_1 and o_2, but is indifferent if he can't have both. We didn't present this operator for the preference annotation in Section 3.2 as we have not encountered such strong conjunction in our corpora. Yet, the following rules are intended to be generic and thus also present the cases associated with this kind of conjunction.

5.2.1.1 Commit

Commit is used to introduce in the CP-net description $\mathcal{DN}$ the variables expressed in the discourse segment π. The rule associated to *Commit(*π*,* $\mathcal{DN}$*)* is composed of seven cases according to the form of π.

a. For $x(P(\pi))$ (x is the value of variable X expressed in $P(\pi)$, for example, *I want* x), we add:

   - $\mathcal{DN} \models x \succ \overline{x}(CPT(X))$.

   Given our description language semantics, this means that any CP-net which satisfies the description $\mathcal{DN}$ contains a preference table $CPT(X)$ with an entry $x \succ \overline{x}$ with at least one instantiation of the variables in $Pa(X)$.

b. For $not\ x(P(\pi))$ (x is an unacceptable outcome, i.e. what the agent does not prefer, for example, *I don't want* x), we add:

- $\mathcal{DN} \models \overline{x} \succ x(CPT(X))$.

c. For $x\&y(P(\pi))$ (the agent prefers to have both x and y and prefers either one if he can't have both), we add:

- $\mathcal{DN} \models y \succ \overline{y}(CPT(Y))$ and
- $\mathcal{DN} \models x \succ \overline{x}(CPT(X))$.

Graphically, this yields the following preference relation (where one way arrows denote preference, two way arrows denote indifference or equal preference, and no arrow means the options are incomparable):

$$\begin{tikzcd}
& xy \arrow[dl] \arrow[dr] & \\
\overline{x}y \arrow[dr] & & x\overline{y} \arrow[dl] \\
& \overline{xy} &
\end{tikzcd}$$

d. For $x\Delta y(P(\pi))$ (the agent prefers to have both x and y but is indifferent if he can't have both), we add:

- $X \in Pa(Y)$ and $\mathcal{DN} \models x : y \succ \overline{y}(CPT(Y))$,
  $\mathcal{DN} \models \overline{x} : y \sim \overline{y}(CPT(Y))$,
- $Y \in Pa(X)$ and $\mathcal{DN} \models y : x \succ \overline{x}(CPT(X))$,
  $\mathcal{DN} \models \overline{x} : y \sim \overline{y}(CPT(Y))$.

Graphically, this yields the following preference relation:

$$\begin{tikzcd}
& xy \arrow[dl] \arrow[d] \arrow[dr] & \\
\overline{x}y \arrow[r, leftrightarrow] & x\overline{y} \arrow[r, leftrightarrow] & \overline{xy}
\end{tikzcd}$$

e. For $x \bigtriangledown y(P(\pi))$ (the agent prefers to have one of x and y satisfied and he is indifferent between the choice of the two outcomes), we add:

- $X \in Pa(Y)$ and $\mathcal{DN} \models x : y \sim \overline{y}(CPT(Y))$,
  $\mathcal{DN} \models \overline{x} : y \succ \overline{y}(CPT(Y))$,
- $Y \in Pa(X)$ and $\mathcal{DN} \models y : x \sim \overline{x}(CPT(X))$,
  $\mathcal{DN} \models \overline{y} : x \succ \overline{x}(CPT(X))$.

This yields the following preference relation:

$$\begin{tikzcd}
\overline{x}y \arrow[r, leftrightarrow] \arrow[dr] & x\overline{y} \arrow[r, leftrightarrow] \arrow[d] & xy \arrow[dl] \\
 & \overline{xy} &
\end{tikzcd}$$

f. For $x \mapsto y(P(\pi))$ (the agent prefers that x is satisfied and if so that y is also satisfied. If x is not satisfied, it is not possible to define preferences on y), we add:

- $\mathcal{DN} \models x \succ \overline{x}(CPT(X))$
- $X \in Pa(Y)$ and $\mathcal{DN} \models x : y \succ \overline{y}(CPT(Y))$.

Note that this description is also produced by *Elab(*π_1*,* π_2*)* below where $X(P(\pi_1))$ and $Y(P(\pi_2))$ (see rule 1). Thus the implication symbol $\mapsto$ is a "shortcut" in that it represents elaborations whose arguments are in the same EDU.

Graphically, this yields the following preference relation (given that we do not have information about the preferences on y if x is not satisfied, $\overline{x}y$ and $\overline{xy}$ are incomparable):

$$\begin{tikzcd}
 & xy \arrow[dl] \arrow[dr] & \\
\overline{x}y & & x\overline{y} \arrow[dl] \\
 & \overline{xy} &
\end{tikzcd}$$

g. For $\phi(P(\pi))$, we apply the previous rules according to the decomposition of ϕ.

In the following sections, we present the rules for each rhetorical relation of the discursive structure. These rules define how to model the preference evolution as the dialogue progresses.

5.2.1.2 IExplanation and Elaboration

IExplanation, Elab, Plan-Elab and Q-Elab introduce dependencies between preferences. For *IExplanation* (π_1, π_2), the preferences expressed in π_2 explain preferences in π_1, like in π_1*: I want to buy a smartphone* π_2*: to not being lost anymore.* For *Elab*(π_1, π_2), a preference in π_1 is elaborated on or developed in π_2, as in π_1*: I want wine.* π_2*: I want white wine. Plan-Elab*(π_1,π_2) means that π_2 describes a plan for achieving the preferences expressed by π_1. With *Q-Elab*(π_1,π_2), we have a similar dependence between preferences, but the second constituent is a question (so often in practice this means preference commitments from π_1 transfer from one agent to another).

1. *IExplanation*(π_1, π_2), *Elab*(π_2, π_1), *Plan-Elab*(π_2, π_1) and *Q-Elab*(π_2, π_1) follow the same two-step rule.

   i Firstly, preference description $\mathcal{DN}$ is updated according to $P(\pi_2)$ by applying *Commit*(π_2, $\mathcal{DN}$), if π_2 expresses a new preference. If not go to step (ii).

   ii. Secondly, description $\mathcal{DN}$ is modified so that each variable in $P(\pi_1)$ depends on each variable in $P(\pi_2)$, i.e. $\forall X \in Var(P(\pi_1)), \forall Y \in Var(P(\pi_2)), Y \in Pa(X)$. If π_1 expresses a preference, the description $\mathcal{DN}$ is enriched according to $P(\pi_1)$. If it does not, then end.

Step (ii) above depends of the form of π_1 and we now give some details concerning each case. To this end, let ϕ denote a preference formula, ϕ' its corresponding boolean (preference) formula and $\overline{\phi'}$ its negation. then, for $\phi = y$, we obtain $\phi' = y$ and $\overline{\phi'} = \overline{y}$; for $\phi = y\Delta z$ and $\phi = y \mapsto z$, we obtain $\phi' = y \wedge z$ and $\overline{\phi'} = \overline{y} \vee \overline{z}$; and for $\phi = y \bigtriangledown z$ and $\phi = y\&z$, we obtain $\phi' = y \vee z$ and $\overline{\phi'} = \overline{y} \wedge \overline{z}$.

a. For $x(P(\pi_i))$ and $\phi(P(\pi_j))$ (The agent explains his preferences on x by ϕ):

   - If $\succ_X$ is not already defined, we add $\mathcal{DN} \models \phi' : x \succ \overline{x}(CPT(X))$. So, if no preferences on X are already defined, ϕ is a reason to prefer x. However, it is not possible to define preferences on x if ϕ is false.
   - If, on the other hand, preferences on X are already defined, the agent prefers x if ϕ is satisfied, and does not modify his preferences otherwise, i.e. $\succ_{X,\phi'} = x \succ \overline{x}$, $\succ_{X,\overline{\phi'}} = \succ_X$.[1]

[1] If we have $\succ_X$ such that z: $\overline{x} \succ x$, $\overline{z}$: $x \succ \overline{x}$, $\succ_{X,\phi'}$ represents preferences defined by $z \wedge \phi'$ and $\overline{z} \wedge \phi'$, whereas $\succ_{X,\overline{\phi'}}$ represents preferences defined by $z \wedge \overline{\phi'}$ and $\overline{z} \wedge \overline{\phi'}$.

For $\phi = y$, if $\succ_X$ is not already defined, we add $\mathcal{DN} \models y : x \succ \overline{x}(CPT(X))$. We obtain the following relation (given that we do not have information on the preference for X if Y is false, $x\overline{y}$ and $\overline{xy}$ are incomparable):

$$\begin{tikzcd}
 & xy \arrow[dl] \arrow[dr] & \\
x\overline{y} & & \overline{x}y \arrow[dl] \\
 & \overline{xy} &
\end{tikzcd}$$

b. For $x\&z(P(\pi_i))$ and $\phi(P(\pi_j))$ (the agent explains his preferences x and z by ϕ):

- if $\succ_X$ is not already defined, we add: $\mathcal{DN} \models \phi'$: $x \succ \overline{x}(CPT(X))$. Otherwise, $\succ_{X,\phi'} = x \succ \overline{x}$, $\succ_{X,\overline{\phi'}} = \succ_X$.
- $CPT(Z)$ is defined as $CPT(X)$ by replacing value for X by values for Z.

For $\phi = y$, if $\succ_X$ and $\succ_Z$ are not already defined, we add:
$\mathcal{DN} \models y$: $x \succ \overline{x}(CPT(X))$.
$\mathcal{DN} \models y$: $z \succ \overline{z}(CPT(Z))$.
We obtain the following preference relation (again, the lack of preference information on X and Z when Y is false yields incomparability among states where Y is false):

$$\begin{tikzcd}
 & xyz \arrow[dl] \arrow[d] \arrow[dr] & \\
\overline{x}yz \arrow[d] \arrow[dr] & x\overline{y}z & xy\overline{z} \arrow[dl] \arrow[d] \\
\overline{xy}z & \overline{x}y\overline{z} \arrow[d] & x\overline{yz} \\
 & \overline{xyz} &
\end{tikzcd}$$

c. For $x\Delta z(P(\pi_i))$ and $\phi(P(\pi_j))$ (the agent explains his preferences on $x\Delta z$ by ϕ: he wants to satisfy X and Z if ϕ is satisfied):

- we have $Z \in Pa(X)$, $X \in Pa(Z)$,
- if $\succ_X$ is not already defined, we add:
  $\mathcal{DN} \models \phi' \wedge z$: $x \succ \overline{x}(CPT(X))$, and
  $\mathcal{DN} \models \phi' \wedge \overline{z}$: $x \sim \overline{x}(CPT(X))$.
  Otherwise, $\succ_{X,\phi',Z} = x \succ \overline{x}$, $\succ_{X,\phi',\overline{Z}} = x \sim \overline{x}$, $\succ_{X,\overline{\phi'},Z} = \succ_{X,\overline{\phi'},\overline{Z}} = \succ_X$.
- $CPT(Z)$ is defined as $CPT(X)$ by replacing value for X by values for Z.

For $\phi = Y$, if $\succ_X$ and $\succ_Z$ are not already defined, we add:
$\mathcal{DN} \models y \wedge z$: $x \succ \overline{x}(CPT(X))$,
$\mathcal{DN} \models y \wedge \overline{z}$: $x \sim \overline{x}(CPT(X))$,
$\mathcal{DN} \models y \wedge x$: $z \succ \overline{z}(CPT(Z))$,
$\mathcal{DN} \models y \wedge \overline{x}$: $z \sim \overline{z}(CPT(Z))$.
We obtain the following preference relation (again, states where Y is false are incomparable):

$$
\begin{tikzcd}
 & xyz \arrow[d] \arrow[ddl] \arrow[ddr] & \\
 & x\overline{y}z & \\
\overline{x}yz \arrow[r, leftrightarrow] \arrow[d] & \overline{x}y\overline{z} \arrow[r, leftrightarrow] \arrow[d] & xy\overline{z} \arrow[d] \\
\overline{xy}z & \overline{xyz} & x\overline{yz}
\end{tikzcd}
$$

d. For $x \bigtriangledown z(P(\pi_i))$ and $\phi(P(\pi_j))$ (the agent explains his preferences on $x \bigtriangledown z$ by ϕ: he wants to satisfy X or Z if ϕ is satisfied):

- we have $Z \in Pa(X)$, $X \in Pa(Z)$,
- if $\succ_X$ is not already defined, we add:
  $\mathcal{DN} \models \phi' \wedge z$: $x \sim \overline{x}(CPT(X))$, and
  $\mathcal{DN} \models \phi' \wedge \overline{z}$: $x \succ \overline{x}(CPT(X))$.
  Otherwise, $\succ_{X,\phi',Z} = x \sim \overline{x}$, $\succ_{X,\phi',\overline{Z}} = x \succ \overline{x}$, $\succ_{X,\overline{\phi'},Z} = \succ_{X,\overline{\phi'},\overline{Z}} = \succ_X$.
- $CPT(Z)$ is defined as $CPT(X)$ by replacing value for X by values for Z.

For $\phi = Y$, if $\succ_X$ and $\succ_Z$ are not already defined, we add:
$\mathcal{DN} \models y \wedge z$: $x \sim \overline{x}(CPT(X))$,
$\mathcal{DN} \models y \wedge \overline{z}$: $x \succ \overline{x}(CPT(X))$,
$\mathcal{DN} \models y \wedge x$: $z \sim \overline{z}(CPT(Z))$,
$\mathcal{DN} \models y \wedge \overline{x}$: $z \succ \overline{z}(CPT(Z))$.
We obtain the following preference relation (again, states where Y is false are incomparable):

$$\begin{tikzcd}
\overline{x}yz \arrow[r, leftrightarrow] \arrow[d] \arrow[ddr] & xyz \arrow[r, leftrightarrow] \arrow[d] & xy\overline{z} \arrow[d] \arrow[ddl] \\
\overline{xy}z & x\overline{y}z & x\overline{yz} \\
& \overline{x}y\overline{z} \arrow[d] & \\
& \overline{xyz} &
\end{tikzcd}$$

e. For $x \mapsto z(P(\pi_1))$ and $\phi(P(\pi_2))$ (the agent explains his preferences on $X \mapsto Z$ by ϕ: he wants to satisfy X then Z if ϕ is satisfied).

- we have $X \in Pa(Z)$,
- if $\succ_X$ is not already defined, we add $\mathcal{DN} \models \phi' : X \succ \overline{X}(CPT(X))$. Otherwise, there is no need to modify $\succ_X$. This is what we call a "*partial elaboration*". Variables that were evoked since preferences on X were introduced are parents of Z but not of X. For example, if an agent commits to a preference for $Monday$ then $Afternoon$, and later in the discourse he commits to $2oclock$, then $Afternoon$ is $2oclock$'s parent but not $Monday$'s.
- If $\succ_Z$ is not already defined, we add $\mathcal{DN} \models \phi' \wedge X : Z \succ \overline{Z}(CPT(Z))$. Otherwise, $\succ_{Z,(\phi' \wedge X)} = Z \succ \overline{Z}$, $\succ_{Z,(\phi' \wedge \overline{X})} = \succ_{Z,(\overline{\phi'} \wedge X)} = \succ_{Z,(\overline{\phi'} \wedge \overline{X})} = \succ_Z$.

For $\phi = Y$, if $\succ_X$ and $\succ_Z$ are not already defined, we add:
$\mathcal{DN} \models y : x \succ \overline{x}(CPT(X))$,
$\mathcal{DN} \models y \wedge x : z \succ \overline{z}(CPT(Z))$.
We obtain the following preference relation (again, states where Y is false are incomparable. Idem for states where X is false since we do not have information about preferences on Z when W is not satisfied):

$$
\begin{tikzcd}
 & xyz \arrow[dl] \arrow[d] \arrow[dr] & \\
\overline{x}yz \arrow[d] & x\overline{y}z & xy\overline{z} \arrow[dl] \arrow[d] \\
\overline{x}\overline{y}z & \overline{x}y\overline{z} \arrow[d] & x\overline{y}\overline{z} \\
 & \overline{x}\overline{y}\overline{z} &
\end{tikzcd}
$$

f. For $\psi(P(\pi_i))$ and $\phi(P(\pi_j))$, we apply the rules 1 according to the decomposition of ψ.

5.2.1.3 Question-answer pair

QAP being a question-answer pair, its influence on preferences depends of the question type.

2. *QAP*$_B$*(*π_1*,* π_2*)* and π_1 is a *yes/no question*. There are two cases, depending on whether B replies *yes* or *no*:

   - When π_2 is *yes*, B's preference descriptions are updated by applying *Commit(Elab*$_B$ *(*π_1*,*π_2*),* $\mathcal{DN}$*)* (and so B's preference description include preferences expressed by π_1 and π_2).
   - When π_2 is *no*, if $P(\pi_1)$ and $P(\pi_2)$ are consistent, then B's preference descriptions are updated by applying *Commit(Elab*$_B$*(*π_1*,*π_2*),* $\mathcal{DN}$*)*, otherwise, they are updated by applying *Commit(Correction*$_B$ *(*π_1*,* π_2*),* $\mathcal{DN}$*)* (see rule 5).

3. *QAP*$_B$*(*π_1*,* π_2*)* and π_1 is a *wh-question*. B's preferences over variables in π_1 and π_2 are exactly the same as the ones defined for a *yes/no question* where the answer is *yes*. Variables in π_2 will refine preferences over variables in π_1. So, B's preference descriptions are updated by applying *Commit(Elab*$_B$*(*π_1*,* π_2*),* $\mathcal{DN}$*)*.

In previous rules, it is relatively clear how to update the preference commitments. However, in some cases it's not clear what the answer in a *QAP* targets: in *Could we meet the 25 in the morning? No, I can't.*, we do not know if *No* is about the 25 and the morning, or only about the morning. So, we define the following rule for managing cases where the ***target*** is unknown:

4. If we know the target, we can change the description of the CP-net. Otherwise, we wait to learn more.

Note that this rule is true if we are *observer* of the dialogue. If we are *actor* (*agent*), we do not know the target, but we must make a choice to continue the dialogue. As an actor, we have to choose one interpretation and if it is not correct, the other actor will correct it thereafter.

5.2.1.4 Correction

Correction and *Plan-Correction* allow a speaker to rectify a prior commitment to preferences. Self-corrections also occur in the corpus: *I could do it on the 27th. No I can not make it on the 27th, sorry I have a seminar.* *Correction* and *Plan-Correction* can have several effects on the preferences. For instance, they can correct preference entries. That is, given *Correction*(π_1, π_2), some variables in $P(\pi_1)$ are replaced by variables in $P(\pi_2)$ (in the self-correction example, every occurrence of 27 in $P(\pi_1)$ is replaced with $\overline{27}$ and vice versa).

5. *Correction*(π_1, π_2) and $X \in Var(P(\pi_1))$ is replaced by the set of variables $\{Y_1, ..., Y_m\} \in Var(P(\pi_2))$. If $Pa(X) = \emptyset$, we add the description $\mathcal{DN} \models y_k \succ \overline{y}_k(CPT(Y_k))$ for all $k \in \{1, \dots, m\}$ and remove $x \succ \overline{x}(CPT(X))$ (or $\overline{x} \succ x(CPT(X))$). Otherwise, we replace every description of $CPT(X)$ by an equivalent description of $CPT(Y_k)$ (where the values of X are replaced by the values of Y_k for all $k \in \{1, \dots m\}$).

The specific target of the correction behaves similarly to the target of a *QAP*. In some cases we don't know the target, in which case we apply rule 4.

Plan-Correction can also lead to the modification of an agent's own plan because of other agent's proposals. In this case it corrects the list of parent variables on which a preference depends. We call that list of variables the *operative variables*. Once the operative variables are changed, *Plan-Correction* can elaborate a plan if some new preferences are expressed. For example, all agents have agreed to meet next week, so in their CP-net description, there is the entry $week1 \succ \overline{week1}$. Then discussion shows that their availabilities are not compatible and one of them says "okay, that week is not going to work". That does not mean the agent prefers $\overline{week1}$ to $week1$ because both agreed on $week1$ as preferable. Rather, $Week1$ has been removed as an operative variable in the following discourse segments. This leads us to the following rule:

6. For *Plan-Correction(π_1, π_2)* which corrects the list of parent variables, the operative variable list becomes the intersection of all $Pa(X)$ where $X \in Var(P(\pi_1))$. We can now apply *Commit(Plan-Elab(π_1,π_2),$\mathcal{DN}$)*, if $P(\pi_2)$ contains some new preferences ϕ. If the CPT affected by a rule has no entry for the current operative variable list $\mathcal{O}$, then the input $\mathcal{O}\colon \phi$ has to be added to $\mathcal{DN}$.

5.2.1.5 Other relations

Continuation, *Contrast* and *Q-Cont* pattern with the rule for *Elab*. *Alternation* patterns with rule 1.b. *Explanation*, *Explanation**, *Result*, *Q-Clar (clarification question)*, *Comment*, *Summary* and *Acknowledgment* either do nothing or have the same effect on preference elicitation as *Elab*. Sometimes, adding these preferences via the *Elab* rule may yield an unsatisfiable CP-net description, because an implicit correction is involved. If an evaluation of the CP-net (see next section) is performed after a processing of one of these rules shows that the CP-net description is not satisfiable, then we apply the rule 5, associated with *Correction*.

5.2.2 From preference descriptions to models

Each dialogue turn adds constraints monotonically to the descriptions of the CP-nets to which the dialogue participants commit. We have interpreted each new declared variable in our rules as independent, which allows us to give a domain independent description of preference elicitation. However, when it comes to evaluating a CP-net description for satisfiability, we need to take into account various axioms about preference (irreflexivity and transitivity), and axioms for the domain of conversation: in our case, temporal designations (Wednesdays are not Tuesdays and so on). This typically adds dependencies among the variables in the description. In the case of the ***Verbmobil*** domain, since the variable *Monday* means essentially "to meet on Monday", *Monday* implies *Meet* , and this must be reflected via a dependency in the CP-net: we must view the variable *Meet* as filling a hidden slot in the variable *Monday* in the preference description, $meet$: $mon \succ \overline{mon}$. This likewise allows us to fill in the negative clauses of the CP-net description: we can now infer that $\overline{meet}\colon \overline{mon} \succ mon$. These axioms also predict certain preference descriptions to be unsatisfiable. For instance, if we have $mon \succ \overline{mon}$, our axioms imply $mon \succ tues$, $mon \succ wed$, etc. At this point we can calculate, *ceteris paribus*, inconsistencies on afternoons and mornings of particular days.

Domain knowledge also allows us to collapse Boolean valued variables that all denote, say, days or times of the day into multiple valued variables. So for instance, our domain independent algorithm from dialogue moves to preference descriptions might yield:

(5.3) meet $\wedge\, \overline{mon} \wedge \overline{tues} \wedge wed$: am $\succ \overline{am}$

Domain knowledge collapses all Boolean variables for distinct days into one variable with values for days to get:

(5.4) meet $\wedge\, wed$: am $\succ$ pm

This leads to a sizeable reduction in the set of variables that are used in the CP-net.

We can test any CP-net description for satisfiability by turning the description formulas into CP-net entries. Our description automatically produces a directed graph over the parent variables. We have to check that the $\succ$ statements form an irreflexive and transitive relation and that each variable introduced into the CP-net has a preference entry consistent given these constraints. If the description does not yield a preference entry for a given variable X, we will add the indifference formula $x \sim \overline{x}$ as the entry. If our CP-net description meets these requirements, this procedure yields a minimal CP-net. Testing for satisfiability is useful in eliciting preferences from several discourse moves like *Explanation*, *Q-Clar* or *Result*, since in the case of unsatisfiability, we will exploit the Correction rule 5 with these moves.

5.2.3 Treatment of an example

We illustrate in this section how our rules work on an example. Since this dialogue was also evaluated by our judges (cf section 5.3), we give where relevant some details on those annotations. The example is as follows:

(5.5) π_1 A: so, I guess we should have another meeting
π_2 A: how long do you think it should be for.
π_3 B: well, I think we have quite a bit to talk about.
π_4 B: maybe, two hours?
π_5 B: how does that sound.
π_6 A: deadly,
π_7 A: but, let us do it anyways.
π_8 B: okay, do you have any time next week?
π_9 B: I have got, afternoons on Tuesday and Thursday.

π_{10} A: I am out of Tuesday Wednesday Thursday,

π_{11} A: so, how about Monday or Friday

Table 5.1 presents the discourse structure associated with the dialogue (5.5). Turn boundaries occur whenever the speaker changes. A Segmented Discourse Representation Structure (SDRS) is associated to each agent representing all his current commitments, from the beginning of the dialogue to the end of that turn. We adopt a convention of indexing the root label of the n^{th} turn, spoken by agent d, as nd. $Relation(\pi_i, \pi_j)$ indicates that a rhetorical relation holds between the EDU π_i and the EDU π_j and $Relation(\pi_i, [\pi_j - \pi_k])$ indicates a relation between the EDU π_i and a complex segment consisting of $\pi_j, \pi_{j+1}, \ldots, \pi_k$.

π_1 provides an atomic preference. We apply the rule a. and so *Commit*$_A(\pi_1, \mathcal{DN}_A)$ adds the following description where M is the variable for having a meeting:

$\mathcal{DN}_A \models meet \succ \overline{meet}(CPT(M))$.

π_2 We have *Q-Elab*(π_1, π_2). A continues to commit to $meet$ in π_2 and no new preferences are introduced by π_2. We apply rule 1, which makes the $P(\pi_2)$ description the same as $P(\pi_1)$'s:

$\mathcal{DN}_A \models meet \succ \overline{meet}(CPT(M))$.

π_3 is linked to π_2 with *QAP*. B accepts A's preference and we apply the rule 3 since π_2 is a *wh*-question. Thus *Commit*$_B$(*Elab*$_B(\pi_2, \pi_3), \mathcal{DN}_B)$ adds the following description:

$\mathcal{DN}_B \models meet \succ \overline{meet}(CPT(M))$.

It is interesting to note that some judges consider that agent's utterance in π_3 indicates a preference towards "talking a long time" while other judges consider, as our method predicts, that this segment does not convey any preference.

π_4 is linked to π_3 by *Q-Elab*. B commits to a new preference. We apply rule 1, rule 1 and then rule 1a. The preference on the hour is now dependent on the preference on meeting; i.e., $\mathcal{DN}_B \models meet : 2h \succ \overline{2h}(CPT(2h))$, where the variable $2h$ means to meet during two hours. We obtain:

$\mathcal{DN}_B \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_B \models meet : 2h \succ \overline{2h}(CPT(2h))$.

π_5 is related to π_4 with the *Q-Cont* relation. We then follow the same rule as the continued relation, namely *Q-Elab*. We apply rule 1 which does not change the CP-net description of B because π_5 does not convey any preference.

Turn	A's SDRS	B's SDRS
1	π_{1A} : *Q-Elab*(π_1, π_2)	$\emptyset$
2	π_{1A}: is the same as in turn 1	π_{2B} : *Q-Elab*$(\pi_1, [\pi_2 - \pi_5]) \wedge$ *QAP*$(\pi_2, [\pi_3 - \pi_5]) \wedge$ *Q-Elab*(π_3, π) π : *Q-Cont*(π_4, π_5)
3	π_{3A} : *Q-Elab*$(\pi_1, [\pi_2 - \pi_7]) \wedge$ *QAP*$(\pi_2, [\pi_3 - \pi_7]) \wedge$ *Q-Elab*$(\pi_3, [\pi_4, \pi_7]) \wedge$ *QAP*(π, π') π : *Q-Cont*(π_4, π_5) π' : *Contrast*(π_6, π_7)	π_{2B}: is the same as in turn 2
4	π_{3A}: is the same as in turn 3	π_{4B} : *Q-Elab*$(\pi_1, [\pi_2 - \pi_9]) \wedge$ *QAP*$(\pi_2, [\pi_3 - \pi_9]) \wedge$ *Q-Elab*$(\pi_3, [\pi_4 - \pi_9]) \wedge$ *QAP*$(\pi, [\pi_6 - \pi_9]) \wedge$ *Q-Elab*(π', π'') π : *Q-Cont*(π_4, π_5) π' : *Contrast*(π_6, π_7) π'' : *Plan-Elab*(π_8, π_9)
5	π_{5A} : *Q-Elab*$(\pi_1, [\pi_2 - \pi_{11}]) \wedge$ *QAP*$(\pi_2, [\pi_3 - \pi_{11}]) \wedge$ *Q-Elab*$(\pi_3, [\pi_4 - \pi_{11}]) \wedge$ *QAP*$(\pi, [\pi_6 - \pi_{11}]) \wedge$ *Q-Elab*$(\pi', [\pi_8 - \pi_{11}]) \wedge$ *QAP*(π'', π''') π : *Q-Cont*(π_4, π_5) π' : *Contrast*(π_6, π_7) π'' : *Plan-Elab*(π_8, π_9) π''' : *Q-Elab*(π_{10}, π_{11})	π_{4B}: is the same as in turn 4

Table 5.1: The discourse structure for a ***Verbmobil*** dialogue.

π_6 is related to π_5 with *QAP* relation. In this case, it's not clear what is the QAP target and so we apply rule 4: we wait to learn more and we do not change A's CP-net description.

All the judges indicated that segments π_5 and π_6 are ambiguous and therefore hesitated to say if they commit to preferences. For example in π_6, do we have a preference for meeting more than 2 hours or less than 2 hours? This indecision is compatible with the predictions of rule 4.

π_7 A accepts B's preference. We apply rule 1 to obtain:

$\mathcal{DN}_A \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_A \models meet : 2h \succ \overline{2h}(CPT(2h))$.

π_8 is linked to π_7 by *Q-Elab*. B introduces a new preference for meeting next week.

We apply rule 1 to obtain the following description where the variable NW means to meet next week:

$\mathcal{DN}_B \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_B \models meet : 2h \succ \overline{2h}(CPT(2h))$,
$\mathcal{DN}_B \models meet \wedge 2h : nxtWk \succ \overline{nxtWk}(CPT(NW))$.

π_9 is linked to π_8 by *Plan-Elab*. π_9 expresses commitments to preference that already involve a CP-net description. B introduces three preferences: one for meeting on Tuesday, the other for meeting on Thursday and given the conjunction of preferences $tues \wedge thur$, one for time afternoon (of Tuesday and Thursday). That is, $((\Diamond(tues) \wedge \Diamond(thur)) \mapsto \mathit{aft})(P(\pi_9))$. We apply the equivalence $(\Diamond o_1 \wedge \Diamond o_2) \leftrightarrow \Diamond(o_1 \vee o_2) \rightarrow o_1 \bigtriangledown o_2$ presented in Section 3.2.2.2 and obtain: $((tues \bigtriangledown thur) \rightarrow \mathit{aft})(P(\pi_9))$.

Then, we apply rules 1b and 1d. The CP-net description of B is thus updated as follows:

$\mathcal{DN}_B \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_B \models meet : 2h \succ \overline{2h}(CPT(2h))$,
$\mathcal{DN}_B \models meet \wedge 2h : nxtWk \succ \overline{nxtWk}(CPT(NW))$,
$\mathcal{DN}_B \models meet \wedge 2h \wedge nxtWk \wedge \overline{tues} : thur \succ \overline{thur}(CPT(Thur))$,
$\mathcal{DN}_B \models meet \wedge 2h \wedge nxtWk \wedge tues : thur \sim \overline{thur}(CPT(Thur))$,
$\mathcal{DN}_B \models meet \wedge 2h \wedge nxtWk \wedge \overline{thur} : tues \succ \overline{tues}(CPT(Tues))$,
$\mathcal{DN}_B \models meet \wedge 2h \wedge nxtWk \wedge thur : tues \sim \overline{tues}(CPT(Tues))$,
$\mathcal{DN}_B \models meet \wedge 2h \wedge nxtWk \wedge (thur \vee tues) : \mathit{aft} \succ \overline{\mathit{aft}}(CPT(\mathit{Aft}))$.

Most judges express here a preference ranking over outcomes. For instance, if B elaborates by adding the preference "I have got Monday morning too" (as it is in the test corpus), some consider the ranking "(Tuesday or Thursday afternoons) $\succ$ (Monday morning) $\succ$ (other days)", while others consider the ranking "(Tuesday or Thursday afternoon) or (Monday morning) $\succ$ (other days)". We did not treat such preference ranking because we think that it does not affect our reasoning.

π_{10} is related to π_9 by ***QAP*** where A answers no to B's question asked in π_8. We apply rule 2 (**no**). Since $\overline{tues}\&\overline{wed}\&\overline{thur}(P(\pi_{10}))$ is not consistent with $((tues \bigtriangledown thur) \rightarrow aft)(P(\pi_9))$, we then apply the rule 5 ***Commit***$_A$ (***Correction***(π_9, π_{10}), $\mathcal{DN}_A$) where $tues$ and $thur$ are respectively replaced by $\overline{tues}$ and $\overline{thur}$ during the commit:

$\mathcal{DN}_A \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_A \models meet : 2h \succ \overline{2h}(CPT(2h))$,
$\mathcal{DN}_A \models meet \wedge 2h : nxtWk \succ \overline{nxtWk}(CPT(NW))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{tues} \succ tues(CPT(Tues))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{thur} \succ thur(CPT(Thur))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{wed} \succ wed(CPT(Wed))$.

π_{11} Finally, π_{11} is linked to π_{10} with ***Q-Elab*** where $mon \bigtriangledown fri(P(\pi_{11}))$. We apply rules 1 and 1b and update A's CP-net description as follows:

$\mathcal{DN}_A \models meet \succ \overline{meet}(CPT(M))$,
$\mathcal{DN}_A \models meet : 2h \succ \overline{2h}(CPT(2h))$,
$\mathcal{DN}_A \models meet \wedge 2h : nxtWk \succ \overline{nxtWk}(CPT(NW))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{tues} \succ tues(CPT(Tues))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{thur} \succ thur(CPT(Thur))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk : \overline{wed} \succ wed(CPT(Wed))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk \wedge \overline{tues} \wedge \overline{thur} \wedge \overline{wed} \wedge \overline{fri} : mon \succ \overline{mon}(CPT(Mon))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk \wedge \overline{tues} \wedge \overline{thur} \wedge \overline{wed} \wedge fri : mon \sim \overline{mon}(CPT(Mon))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk \wedge \overline{tues} \wedge \overline{thur} \wedge \overline{wed} \wedge \overline{mon} : fri \succ \overline{fri}(CPT(Fri))$,
$\mathcal{DN}_A \models meet \wedge 2h \wedge nxtWk \wedge \overline{tues} \wedge \overline{thur} \wedge \overline{wed} \wedge mon : fri \sim \overline{fri}(CPT(Fri))$.

5.3 Evaluation of the proposed method

We evaluate our method by testing it against the judgements of three annotators on three randomly chosen unseen test dialogues from ***Verbmobil***. The test corpus

	Our algorithm	J1	J2	J3	% of EDU with preferences
Our algorithm		(83, 4, 13)	(91, 0, 9)	(91, 0, 9)	76%
J1	(83, 13, 4)		(85, 7, 8)	(91, 4, 5)	80%
J2	(91, 9, 0)	(85, 8, 7)		(92, 4, 4)	86%
J3	(91, 9, 0)	(91, 5, 4)	(92, 4, 4)		84%

Table 5.2: Evaluation for preference extraction from each EDU.

	Our algorithm	J1	J2	J3
Our algorithm		(85, 71)	(96, 100)	(93, 86)
J1	(85, 71)		(89, 71)	(91, 86)
J2	(96, 100)	(89, 71)		(98, 86)
J3	(93, 86)	(91, 86)	(98, 86)	

Table 5.3: Evaluation for preference evolution through dialogue.

contains 75 EDUs and the proportion of discourse relations is the same as in the corpus overall. The three annotators were *naive* in the sense that they were not familiar with preference representations and preference reasoning strategies. For each dialogue segment, we checked if the judges had the same intuitions that we did on: (i) how commitments to preferences are extracted from EDUs, and (ii) how preferences evolve through dialogue exchange.

The judges were given a manual with all the instructions and definitions needed to make the annotations. For example, the manual defined preference to be "a notion of comparison between one thing at least one other". The manual also instructs annotators to label each EDU with the following four bits of information: (1) preferences (if any) expressed in the EDU; (2) dependencies between preferences expressed in the EDU; (3) dependencies between preferences in the current EDU and previous ones; and (4) preference evolution (namely, the appearance of a new factor that affects preferred outcomes, update to preferences over values for an existing factor, and so on). For each of these four components, example dialogues were given for each type of decision they would need to make, and instructions were given on the format in which to code their judgements. Section 5.2.3 shows an example of an annotated dialogue.

Table 5.2 presents results of the evaluation of (i). For each EDU, we asked the

annotator to list the preferences expressed in the EDU and we compared the preferences extracted by each judge with those extracted by our algorithm. The triple (a, b, c) respectively indicates the proportion of common preferences (two preference sets Γ_i and Γ_j are common if ($\Gamma_i = \Gamma_j$) or ($\exists x \in \Gamma_i, y \in \Gamma_j, x \to y$)—for example, the preference *meetBefore2* $\succ$ *meetAt2* implies $\overline{\textit{meetAt2}} \succ$ *meetAt2*), the proportion of preferences that one judge extracts and the other judge or our algorithm misses and the proportion of preferences missed by one judge and extracted by the other judge or by our algorithm. The average annotator-algorithm agreement (AAA) is 75.6% and the average inter-annotator agreement (IAA) is 77.9%; this shows that our method for extracting preferences from EDUs is reliable.
The evaluation (ii) proceeds as follows. For each EDU, we ask the judge if the segment introduces new preferences or if it updates, corrects or deletes preferences commited in previous turns. As in (i), judges have to justify their choices. Table 5.3 presents the preliminary results where the couple (a,b) indicates respectively the proportion of common elaborations (preference updates or new preferences) and the proportion of common corrections. Since elaboration is also applied in case of other discourse relations (e.g., *Q-Elab*), the measure a evaluates the rules 1, 1, 2 (**yes**) and 3. Similarly, the measure b evaluates the rules 2 (**no**), 5 and 6. We obtain AAA=91% IAA=92.7% for elaboration and AAA=85.7% IAA=81% for correction.

The evaluation of this dialogue also reveals to what extent naive annotators reason with binary (*Monday preferred to not Monday*) or multi-valued variables (*Monday preferred to Tuesday*). Most judges use multi-valued variables to express the preference extracted from an EDU, and the way in which our method exploits domain knowledge to yield the minimal CP-net satisfying the description reflects this. In addition, some judges use a small set of variables (for example the variable *time of meeting* that groups together the notion of week, day, hours, etc.) while others use a distinct variable for each preference.

Finally, we also noticed that judges do not describe the same preference dependencies. For example, in:

(5.6) We could have lunch together and then have the meeting from one to three?

some consider that the preference on having lunch is independent from the preference on the meeting (in this case, they consider that the preference on the period one to three is independent from the preference on meeting) while others consider that the two preferences are dependent.

5.4 Conclusion

In this chapter, we have proposed a compositional method to elicit preferences from dialogues consisting of a domain-independent algorithm for the construction of a partial CP-net description of preferences, followed by a domain-specific method to identify the minimal CP-net satisfying the partial description and domain constraints. The method supports qualitative and partial information about preferences, with CP-nets benefiting from linear algorithms for computing the optimal outcome from a set of preferences and their dependencies. The average annotator-algorithm agreement and the average inter-annotator agreement show that our method is reliable.

The need to compute intentions from partially defined preferences is crucial in dialogue, since preferences are acquired and change through dialogue exchange. This work partially confirms that CP-nets have a certain naturalness, as the map from dialogue moves to preferences using the CP-net formalism is relatively intuitive and we present in the following chapter how the method is well suited to predict trades in the game *The Settlers of Catan*.

Chapter 6

Preferences and strategy prediction

6.1 Dialogue act and resource prediction **144**
6.1.1 Identifying dialogue acts 145
6.1.2 Finding resource text spans 146
6.1.3 Recognizing the type of resources 147
6.2 Predicting players' strategic actions **148**
6.2.1 CP-Nets in our ***Settlers*** corpus 149
6.2.2 Modelling players' preferences 149
6.2.3 Example 151
6.2.4 Evaluation and results 153
6.3 Conclusion **156**

In this chapter, we rely on the results obtained in Chapters 4 and 5 to predict trades in the win-lose game *The Settlers of Catan*. We estimate the preferences of EDUs automatically and exploit the conversation to dynamically construct a partial model of each players preferences in dialogues from our ***Settlers*** corpus of negotiation dialogues. This partial model of preferences in turn yields equilibrium trading moves via principles from game theory.

Modelling player behaviour in real-time strategy games is a growing research area in AI. These models can be used to identify common strategic states and decision points, or discover novel strategies as they emerge. They can also predict an opponent's future actions and so help players to optimize their strategies. Various techniques have been proposed to automatically predict strategic actions. For example, Schadd et al. (2007) develop a hierarchical opponent model in the game

Spring, Dereszynski et al. (2011) learn and reason about strategic behaviour in *StarCraft* using hidden Markov models, and Amato and Shani (2010) use reinforcement learning to acquire a policy for switching among high-level strategies in *Civilization IV*.

By comparison, we propose a novel approach for predicting strategic action that's based on the *symbolically* formalized preferences that each agent commits to in spontaneous conversation. In doing so, our approach deals with imperfect information by exploiting the agents' declared preferences. By predicting what bargain (if any) will take place, we are able to verify the correctness of our preference descriptions. Our task is a subtask of learning a strategy over an entire game space, but our approach yields good predictive results on relatively little data—an advantage of exploiting CP-nets and the symbolic rules that guide their evolution from observable evidence.

We design a model that maps what people say in the win lose game *The Settlers of Catan* into a prediction of exactly which players, if any, trade with each other, and exactly what resources they exchange. We use both statistics and logic. Specifically, we use the ***Settlers*** corpus to learn classifiers that map each utterance to its speech act and to other acts that are pertinent to bargaining. And we develop a symbolic algorithm that, from the classifiers' output, dynamically constructs a model of each player's preferences as the conversation proceeds (for instance, the preference to receive a certain resource, or to accept a certain trade). This preference model is based on CP-nets (see Section 1.3.3), a logic of preferences for which algorithms for computing equilibrium strategies exist. We adapt those algorithms to predict the trades that are executed in the game.

Section 6.1 introduces our method to construct the agents' preferences from the dialogues. We use this in Section 6.2 to predict whether a trade is executed as a result of the players' negotiations, and if so we predict who took part in the trade, and what they exchanged. Our method shows promising results, beating baselines that don't adequately track or reason about preferences.

6.1 Dialogue act and resource prediction

Most work on dialogue act modelling focuses on spoken dialogue (Stolcke et al., 2000; Fernández et al., 2005; Keizer et al., 2002). But live chats introduce specific complications (Kim et al., 2012): ill-formed data, abbreviations and acronyms, emotional indicators and entanglement (especially for multi-party chat). Among related work in this emerging field, Joty et al. (2011) use unsupervised learning to model dialogue acts in Twitter, Ivanovic (2008) and Kim et al. (2010) analyze one-to-one online chat in a customer service domain, and Wu et al. (2002) and Kim et al. (2012) predict dialogue acts in a multi-party setting. We used a similar classifier to predict dialogue acts as the one reported in (Kim et al., 2012) and evaluation yields similar results.

In this section, we propose an approach to dialogue act identification in online chat that aims to predict strategic actions like bargaining. So compared to (Sidner, 1994) and DAMSL (Core and Allen, 1997), our domain level annotation is much more detailed: we not only predict moves like *Accept* but also features like the *Givable* and *Receivable* resources. Our general speech act typology of EDUs lacks intentional descriptions of speech acts, however. This reflects a conscious choice to specify the semantics of each act purely by the public commitments made to offer or to receive goods.

Before to present how to predict the executed trades from the dialogues in the following section, we detail here the pretreatment in three sub-tasks: (1) automatically identifying each EDU's dialogue act (i.e., *Offer*, *Counteroffer*, etc.); (2) detecting the EDU's resources; and (3) specifying the attributes of those resources (i.e., *Givable*, *Receivable*, etc.).

6.1.1 Identifying dialogue acts

We recall from Section 3.1.2.2 the five types of dialogue act in our corpus: *Offer*, *Counteroffer* (which may be a reply to an offer or a specification of an offer the speaker has already made), *Accept* or *Refusal* (of an offer addressed to the emitter), and *Other* (which may be comment on strategic moves in the game or moves not directly pertinent to bargaining).

As is well established, one EDU's dialogue act depends on previous dialogue acts (Stolcke et al., 2000). In our corpus, *Accept* or *Reject* frequently follow *Offer* and *Counteroffer*. Since labeling is sequential, we use Conditional Random Fields (CRFs) to learn dialogue acts. CRFs have been shown to yield better results in dialogue act classification on online chat than HMM-SVN and Naive Bayes (Kim et al., 2012).

We use three types of features: lexical, syntactic and semantic. And we exploit them as unigrams and bigrams: unigrams associate the value of the feature with the current output class (level 0); bigrams take account of the value of the feature associated with a combination of the current output class and previous output class (level -1). 6 features were used exclusively as unigrams: the EDU's position in the dialogue, its first and last words, its subject lemma, a boolean feature to indicate if the current speaker is the one that initiates the dialogue and the position of the speaker's first turn in the dialogue.

We have 15 unigram and bigram features (at levels 0 and -1), as well as templates that combine feature values for the two levels. These include 14 boolean features that indicate if the EDU contains: bargaining verbs (e.g. *trade, offer*), references to another player (e.g. *you*), resource tokens as encoded in a task dedicated lexicon (such as *wheat, clay*), quantifiers (e.g. *one, none*), anaphoric pronouns, occurrences of "for" prepositional phrases (e.g. *wheat for clay*), acceptance words (*OK*), negation words, emoticons, opinion words (taken from (Benamara et al., 2011)), words of politeness, exclamation marks, questions, and finally whether the EDU's speaker has talked previously in the dialogue. The last feature gives the EDU speaker lemma. In addition 3 unigram and bigram boolean indicate whether the current EDU contains the most frequent tokens, couple of tokens and syntactic patterns in our corpus. Finally, we have 2 composed bigram features that encode whether the EDU contains an acceptance or refusal word, given that the previous EDU is a question.

We use the CRF++ tool[1] to assign sequential tags of dialogue acts within a negotiation dialogue. Our data consists of 511 dialogues, or 2741 EDUs. Each EDU is associated with a dialogue act resulting in 410 *Offer*, 197 *Counteroffer*, 179 *Accept*, 398 *Refusal* and 1557 *Other*.

We use 10-fold cross-validation to evaluate our model, computing precision, recall and F-score for each class and global accuracy from the total number of true positives, false positives, false negatives and true negatives obtained by summing over all fold decisions. The results (in percents) are given in Table 6.1 (MaF is the average of F-scores of all the classes). Our model significantly outperforms the frequency-based baseline (MaF=14.5; Accuracy=56.8), with the best F-score achieved for *Other*. The least good results are for the two least frequent classes in our data. In addition to the frequency problem, the lower score for *Counteroffer* is mainly due to the model confusing it with *Offer*. Errors in the *Accept* class were often due to misspelling or to chat style conversation; e.g., *kk*, *yup*.

6.1.2 Finding resource text spans

Since the resource vocabulary in *The Settlers of Catan* is a closed set composed of words denoting specific resources (e.g., *clay*, *wood*) and their synonyms (*brick*), we use a simple rule to detect them: a Noun Phrase (NP) is a resource text span if and only if it contains a lemma from our resource lexicon. A closed set resource vocabulary is common to many different types of negotiation dialogues. We used the Stanford parser (Klein and Manning, 2003) to obtain the NPs: there are 4361 NPs, where (by the gold standard annotations) 21% are resources and 79% are not.

[1] http://crfpp.googlecode.com/svn/trunk/doc/index.html

Dialogue act	Precision	Recall	F-score
Other	87.4	93.1	90.1
Offer	80.0	81.0	80.5
Counterof.	64.8	53.3	58.5
Accept	65.1	53.1	58.5
Refusal	81.7	73.9	77.6
Macro-averaged F-score (MaF)			**73.0**
Accuracy			**83.0**

Table 6.1: Results for dialogue act classification.

We obtain an F-score of 96.9% and accuracy of 97.9%, clearly beating both the frequency and random baselines for this task.

6.1.3 Recognizing the type of resources

Recall that each resource within an EDU can be the value of four types of attributes: *Givable, Receivable, Not Givable* or *Not Receivable*. We predict these attributes using CRFs with the following features. 8 features are used as unigram at the current and the previous EDU level: the speaker, the EDU's subject, the dialogue act, and (if present) the lemma of a bargaining verb, and 4 boolean features indicate if the EDU contains an opinion word, a reference to another speaker, if the resource comes after a "for" and if it contains a refusal word. These features also serve as bigrams at the current EDU level. Additionally, we have a set of unigram and bigram boolean features that indicate if the current EDU contains the most frequent verbs in the corpus. And finally, we use a feature that encodes the combination subject/bargaining verb in the current EDU.

We used CRF++ to implement our classifier. Our corpus data consists of 1077 Resources, split into 510 *Receivable*, 432 *Givable*, 116 *Not Givable* and 19 *Not Receivable*. We use again 10-fold cross-validation to evaluate our model and compute the results by summing over all fold decisions. We present them (in percents) in Table 6.2. They beat the frequency-based baseline (MaF=16.1; Accuracy=47.4), although performance on the *Not Receivable* class is poor probably due to its low frequency in the data.

Ambiguities make this task challenging. For instance, *anyone wheat for clay?* can mean that the speaker wants to receive wheat and give clay or the opposite, and resolving which meaning is intended involves reasoning not only with the previous and/or the following EDU, but also sometimes EDUs with long distance attachments, which are not supported by our classifier and require a full discourse parser.

Res. type	Precision	Recall	F-score
Receivable	66.8	71.4	69.0
Givable	62.6	59.7	61.1
Not Giv.	88.1	89.7	88.9
Not Rec.	0	0	0
Macro-averaged F-score (MaF)			**54.8**
Accuracy			**67.4**

Table 6.2: Results for resource type classification.

6.2 Predicting players' strategic actions

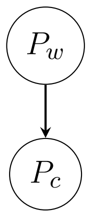
We aim to capture the evolution of commitments to certain preferences as the dialogue proceeds so as to predict the agents' bargaining behaviour. In other words, we wish to predict which of the 61 possible trade actions is executed at the end of each dialogue. The possible trades vary over which partner the player whose turn it is trades with (3 options in a 4 player game), the resources exchanged (assuming each partner gives one type of resource and receives another type yields $5 \times 4 = 20$ possibilities), or there is no trade; i.e., $(3 \times 20) + 1 = 61$ possible actions in the hypothesis space (we predict the types of resources that are exchanged, but not their quantity).

We predict the executed action by identifying the equilibrium trade that's entailed by the model of the players' preferences, which in turn we construct dynamically. We use the attributes of resources in the EDUs (*Givable*, etc.) to identify the preference that a speaker conveys in the EDU, and we use the dialogue acts (*Offer*, *Accept*, etc.) to update a model of the preferences expressed so far in the dialogue with this new preference (see Section 6.2.2). Our model of preferences consists of a set of partial CP-nets (Boutilier et al., 2004), one for each player (see Section 6.2.1 for details). The resulting CP-nets are then used to automatically infer the executed trading action (if any), via well-understood principles from game theory for identifying rational behaviour (Bonzon, 2007).

6.2.1 CP-Nets in our *Settlers* corpus

Let's consider how CP-nets are constructed from the dialogues. In the ***Settlers*** corpus, preferences involve a quadruplet *(o, a, <r,q>)* where: *o* is the preference owner, *a* is the addressee, *r* is the resource and *q* is its quantity (see Section 3.2.3.2). So each variable in the CP-nets we construct is such a quadruplet, and for each variable the possibles values are *Givable* (Giv), *Not Givable* ($\overline{Giv}$), *Receivable* (Rcv) and *Not Receivable* ($\overline{Rcv}$).

For example, the utterance *Anyone want to give me a wheat for a clay?* expresses two preferences: one for receiving wheat, represented by the variable P_w = (A,All,<wheat,1>); and given this preference, another for giving clay, represented by P_c = (A,All,<clay,1>) (where A is the name of the speaker). The corresponding CP-Net is Figure 6.1.



$\text{CPT}(P_w) = \quad Rcv \succ \overline{Rcv}$

$\text{CPT}(P_c) = \quad Rcv\_P_w : Giv \succ \overline{Giv}$

Figure 6.1: A CP-net example from the ***Settlers*** corpus.

6.2.2 Modelling players' preferences

As stated above, we first automatically acquire a CP-net from each EDU by using the EDU's dialogue act and the attributes (*Givable*, etc.) of its resources. We then apply the rules presented in Chapter 5 to dynamically construct a preference model of the dialogue overall. We do not directly use the discourse structure since no automatic parser exists for our ***Settlers*** corpus.[2] Thus in order to provide a complete automatic method, we use an equivalence between the rhetorical relations used in Chapter 5 and the dialogue acts we can automatically extract (see Section 6.1.1). We present below the equivalence for each kind of dialogue act (in the remainder of the section, π_i stands for EDU ID i):

Offers. Because an *Offer* may specify or refine an existing preference or offer, we need to model how the preferences expressed in an EDU that's an *Offer* updates

[2]Note that a parser exists for the ***Verbmobil*** corpus (Baldridge and Lascarides, 2005b).

the prior declared preferences. Therefore, while our annotations treat *Offer* as a property of EDUs, we treat them here as *binary relations*: *Offer*(π_1, π_2), where the second term, π_2, is the actual EDU whose dialogue act is *Offer* and π_1 is the set of EDUs occurring between π_2 and the last EDU uttered by the same speaker. *Offers* then have a similar effect on the CP-net as the coherence relation *Elaboration* presented in Section 5.2. That is, to automatically update the CP-net constructed so far with a current EDU that's an *Offer*, the two step rule for *Offer*(π_1, π_2) is:

1. to update the speaker's CP-net according to the preferences expressed in π_1, and
2. if π_2 expresses preferences, to enrich the CP-net with these new preferences so that each variable in π_2 depends on each variable in π_1.

Counteroffers. *Counteroffers* specify or modify the terms of a previous *Offer* or *Counteroffer*. Their purpose is to give new information to refine the negotiation. So like *Offers* they must also receive a contextually dependent interpretation. The rule is quite similar to that for *Offer*; however, *Counteroffer* can modify or *correct* elements in a previously introduced offer. So for *Counteroffer*(π_1, π_2), the rule is:

1. to *partially* update the speaker's CP-net according to the preferences expressed in π_1 which do not have the same Resource type (*Givable, Receivable*) than the ones in π_2.
2. same as step 2 *Offer* rule.

Accepts and Refusals. *Accepts* or *Refusals* are answers to *Offers* and *Counteroffers*. Thus, they behave like question answer pairs (*QAPs*) presented in Section 5.2. Because we are not doing full discourse parsing, we once again approximate its effects by making *Accepts* and *Refusals* respond to the set of EDUs between the current EDU and the speaker's last turn.

1. *Accepts* are positive responses to *Offers* or *Counteroffers* and are *de facto* similar to *QAP*(π_1, π_2) where π_2 is *Yes*. Thus, the rule is, as for the *Offer*, to update and enrich the CP-net.
2. *Refusals* are instead negative responses and behave like *QAP*(π_1, π_2) where π_2 is *No*. For *Refusal*(π_1, π_2), there is no update of the preferences expressed in π_1. Instead, we enrich the CP-net with the *Non Givable* and *Non Receivable* information obtained from the negation of the preferences expressed in the previous *Offer* or *Counteroffer*. We then enrich the CP-net based on any new preferences expressed in π_2. If there is a conflict between the value of

a variable to be updated and the current value in the CP-net, we apply the *Correction* rule: all occurrences of the old value is replaced by the new value in π_2.

Other. This category pertains to content that does not directly relate to trading in the game, and so we choose to ignore resources expressed in the EDUs with this dialogue act.

At the end of the negotiation dialogue, to predict exactly what trade is executed (if any), the method checks if there are complete and reciprocal preferences expressed in the CP-nets that respectively represent the declared preferences of two agents A and B. This is done in two steps. First, we use the logic of CP-nets to determine each agent's best outcome $bestO_A$ and $bestO_B$ from their respective CP-nets (we'll discuss how shortly). Secondly, we compare these best outcomes: if they correspond to the same trade, we predict that this trade was executed; if not, we predict no trade is executed. Specifically, $bestO_A$ (resp. $bestO_B$) corresponds to a preference for receiving a resource r_1 from an agent B (or from all the agents indifferently) and for giving a resource r_2 to this (or these) agent(s). We predict that A gives B r_2 and B gives A r_1 if and only if: $bestO_A = Rcv(A, B, r_1) \wedge Giv(A, B, r_2)$ and $bestO_B = Rcv(B, A, r_2) \wedge Giv(B, A, r_1)$.

The first step—computing each agent's best outcome from his CP-net—can be found in linear time using the *forward sweep algorithm* (Boutilier et al., 2004): sweep through the CP-net's graph from top to bottom, instantiating each variable with its preferred value, given the values that are (already) assigned to its parents. This algorithm is *sound* with respect to the logic of CP-nets.

6.2.3 Example

We apply this method to construct CP-nets and determine the executed trade to the negotiation dialogue in Table 6.3, using the gold standard annotations as input.

π_1 The EDU is an *Offer*, so Rainbow's CP-net is updated according to π_1's content.

CPT(R,All,<clay,?>) = $Rcv \succ \overline{Rcv}$

π_2 It's a *Refusal*, so we update inca's CP-net with the negation of the preferences expressed in Rainbow's offer.

CPT(I,R,<clay,?>) = $\overline{Giv} \succ Giv$

π_3 Idem for ariachiba.

CPT(A,R,<clay,?>) = $\overline{Giv} \succ Giv$

ID	Dialogue Act	Text	Speaker	Addressee	Resource
1	Offer	i need clay, any1 have?	Rainbow	All	Receivable (clay, ?)
2	Refusal	Nope, sorry	inca	Rainbow	
3	Refusal	Not at the moment.	ariachiba	Rainbow	
4	Refusal	need mine sorry	Kittles	Rainbow	Not givable (Anaphoric, ?) Anaphora Link:(mine , clay)
5	Offer	no one has ore to giv?	Rainbow	All	Receivable (ore, ?)
6	Accept	oh yeah me	Kittles	Rainbow	
7	Counteroffer	ore for wheat again?	Kittles	Rainbow	Givable (ore, ?) Receivable (wheat, ?)
8	Accept	ya	Rainbow	Kittles	
9	Accept	ok	Kittles	Rainbow	

Table 6.3: The negotiation annotation for a ***Settlers*** dialogue.

π_4 Idem for Kittles where the preferences expressed in this EDU are redundant with the negation of the offer preferences.

$$\mathit{CPT(K,R,<clay,?>)} = \overline{Giv} \succ Giv$$

π_5 The EDU is an *Offer*, so Rainbow's CP-net is first updated according to previous EDUs (π_2 to π_4 until his last speaking), then according to the content of π_5. So we obtain:

$$\mathit{CPT(R,All,<clay,?>)} = Rcv \succ \overline{Rcv} \qquad \textbf{\textit{(inactive)}}$$
$$\mathit{CPT(R,I,<clay,?>)} = \overline{Rcv} \succ Rcv$$
$$\mathit{CPT(R,A,<clay,?>)} = \overline{Rcv} \succ Rcv$$
$$\mathit{CPT(R,K,<clay,?>)} = \overline{Rcv} \succ Rcv$$
$$\mathit{CPT(R,All,<ore,?>)} = \overline{Rcv}\mathit{(R,I,<clay,?>)} \wedge \overline{Rcv}\mathit{(R,A,\ <clay,?>)} \wedge \overline{Rcv}\mathit{(R,K,<clay,?>)}\text{: } Rcv \succ \overline{Rcv}$$

The introduction of the new preference to receive ore conflicts with the prior one for receiving clay. So the method adds to the associated CPT the label "inactive" as shown, to indicate that this is older and should be ignored if the preference about ore is satisfied.

π_6 The EDU is an *Accept*, so Kittles's CP-net is updated according to previous EDUs (only π_5).[3]

$$\mathit{CPT(K,R,<ore,?>)} = \overline{Giv}\mathit{(K,R,<clay,?>)}\text{: } Giv \succ \overline{Giv}$$

π_7 The EDU is a *Counteroffer*. Since she is the last speaker, her CP-net gets updated only according to the content of the current EDU, to obtain:

[3] In the following CP-nets, we do not copy the inactive CPTs and CPTs about *Not Givable* or *Not Receivable* resources.

$CPT(K,R,<ore,?>) = \overline{Giv}(K,R,<clay,?>): Giv \succ \overline{Giv}$
$CPT(K,R,<wheat,?>) = \overline{Giv}(K,R,<clay,?>) \wedge Giv(K,R, <ore,?>): Rcv \succ \overline{Rcv}$

π_8 The EDU is an *Accept*, so Rainbow's CP-net is updated according to previous EDUs (π_6 and π_7):

$CPT(R,K,<ore,?>) = \overline{Rcv}(R,I,<clay,?>) \wedge \overline{Rcv}(R,A, <clay,?>) \wedge \overline{Rcv}(R,K,<clay,?>): Rcv \succ \overline{Rcv}$
$CPT(R,K,<wheat,?>) = \overline{Rcv}(R,I,<clay,?>) \wedge \overline{Rcv}(R,A, <clay,?>) \wedge \overline{Rcv}(R,K,<clay,?>) \wedge Rcv(R,K,<ore,?>): Giv \succ \overline{Giv}$

π_9 The EDU is an *Accept* but there is nothing new to update.

At the end of the dialogue, these agents' CP-nets (correctly) predict that Kittles gave ore to Rainbow in exchange for wheat.

6.2.4 Evaluation and results

We compare our model against four baselines. Since none of these baselines support reasoning about equilibrium moves, they all rely on the presence of an *Accept* act to predict there was a trade, and its absence to predict there wasn't. The baselines differ, however, in how they identify the trading partners and resources in an executed trade.

The first baseline predicts a trade according to the first *Offer* and the last person to *Accept*, and if the *Offer* doesn't specify one of the resources then it is chosen randomly (similar random choices complete all partial predictions in all the models we consider here): e.g., for Table 6.3 this would predict that Kittles gave clay to Rainbow (which is incorrect) in exchange for something that's chosen randomly (which will probably be incorrect).

The second baseline uses the last *Offer* and the last person to *Accept*: e.g., for Table 6.3 this predicts that Kittles gave ore to Rainbow (correct) for something random (probably incorrect).

The third baseline uses the last *Offer* or *Counteroffer*, whichever is latest, and the last person to *Accept*: e.g., for Table 6.3 this correctly predicts that Kittles gave ore to Rainbow in exchange for wheat.

And the fourth baseline, uses *default unification* between the prior *Offers* or *Counteroffers* and the current one to resolve any of the current offer's elided parts and to replace specific values in prior offers with conflicting specific values in the current offer (Ehlen and Johnston, 2013). One then takes the executed trade to be the result of this unification process at the point where the last *Accept* occurs. This makes the same predictions as the third baseline for Table 6.3, but outperforms it in the corpus example (6.1) by predicting the correct and *complete* trade (i.e., Rainbow gave Kittles sheep for wheat, rather than for something random):

(6.1)	Rainbow:	i need clay ore or wheat
	Kittles:	i got wheat
	Rainbow:	i cn giv sheep
	Kittles:	ok

We performed the evaluation on the data presented in Sections 3.1.2: 254 dialogues in total since we ignore dialogues that contain only *Other*s. 90 of these dialogues end with a trade being executed and 2 of them end with 2 trades. A random baseline would give 1.6% accuracy (given the 61 possible trading actions) and a frequency baseline (always choose no trade) gives 64.1% accuracy. Table 6.4 presents the results.

1^{st} baseline: first *Offer*/last *Accept*					
TP	FP	FN	TN	WP	Accuracy
24	14	30	150	38	**68.0**
2^{nd} baseline: last *Offer*/last *Accept*					
TP	FP	FN	TN	WP	Accuracy
29	6	32	158	31	**73.0**
3^{rd} baseline: last *(Counter)Offer*/last *Accept*					
TP	FP	FN	TN	WP	Accuracy
39	4	23	160	30	**77.7**
4^{th} baseline: *default unification*					
TP	FP	FN	TN	WP	Accuracy
64	4	23	160	5	**87.5**
Our method					
TP	FP	FN	TN	WP	Accuracy
75	4	15	160	2	**91.8**

Table 6.4: Results for trade prediction.
TP, FP, FN, TN and WP are the True and False Positives, False and True Negatives and Wrong Positives.

The accuracy of all the models is calculated from the gold standard labels rather than the classifiers' predicted labels from Section 6.1, so that we can compare the models in isolation of the classifiers' errors. McNemar's test shows that our model significantly outperforms all the baselines (see Appendix A.4). A predicted trade counts as correct only if it specifies the right participants (addressee and owner) and the correct type of resources offered and received (we ignore their quantity). True Positives (TP) are thus examples where the model correctly predicts not only that a trade happened, but also the correct partners and resources; Wrong Positives (WP), on the other hand, constitute a correct prediction that there was a trade but errors on the partners and/or resources involved (so WP undermine accuracy). True Negatives (TN) are examples where the model correctly predicts there was no trade (so TP and TN contribute to accuracy). False Positives (FP) and False Negatives (FN) are respectively incorrect predictions that there was a trade, or that there was no trade.

It does not appear in Table 6.4, but the first three baselines tend to predict *incomplete* information about the trade even when what they do predict is correct: that is, they predict the correct addressee and the owner but resort to random choice for a resource that's missing from the *Offer* or *Counteroffer* that predicts which trade occurred. For the first baseline 34 examples are like this; for the second and third baselines it's 32. In contrast, this problem occurs only once with the fourth baseline, and all the trades predicted by our method are complete, making random choice unnecessary. Moreover, the first three baselines often make incorrect predictions about the addressee or resources exchanged because in contrast to our model and the fourth baseline, they don't track how potential trades *evolve* through a *sequence* of offers and counteroffers.

Even if the fourth baseline, which uses default unification to track the content of the current offer, is smart and gives good results, it has statistically significant lower accuracy than our model. One major problem with the fourth baseline is that, in contrast to our model, it does not track each player's *attitude* towards the current offer. Instead, like all our baselines, it relies on the presence of an *Accept* act to predict that there's a trade. But several corpus examples are like (6.2), in which a trade is executed but there's no *Accept* act, thus yielding a False Negative (FN) for all four baselines. We also tried a baseline that doesn't rely on the presence of an *Accept* act, but rather predicts a trade whenever default unification yields a complete offer. It performed worse than the fourth baseline.

(6.2)	Joel:	anyone have sheep or wheat
	Cardlinger:	neither :(
	Joel:	will give clay or ore
	Euan:	not just now
	Jon:	got a wheat for a clay

(*Joel gives clay to Jon and receives wheat*)

So overall, our analysis shows that using CP-nets significantly outperforms all baselines that don't model how preferences evolve in the dialogue, and error analysis yields evidence that our model outperforms the fourth baseline because our model supports *reasoning* about player preferences, rational behaviour and equilibrium strategies.

Table 6.5 presents the results for the end to end evaluation, where trade predictions are made from the classifiers' output from Section 6.1 rather than the gold standard labels. As expected, performance decreases due to the classifiers' errors, mainly on the type of resources (*Givable*, etc.). But our method still significantly outperforms all the baselines with an accuracy of 73.4% when the baselines obtain values between 60.9% and 68.4%.

4th baseline: *default unification*					
TP	FP	FN	TN	WP	Accuracy
23	12	37	152	32	**68.4**
Our method					
TP	FP	FN	TN	WP	Accuracy
34	10	43	154	15	**73.4**

Table 6.5: Results for the end to end trade prediction.

6.3 Conclusion

In this chapter, we proposed a linguistic approach to strategy prediction in spontaneous conversation that exploits dialogue acts to build a partial model of the agents' declared preferences. Our method tracks how preferences evolve during the dialogue, and we exploit this to logically infer their bargaining behaviour, i.e. what resources, if any, are exchanged, and by whom.

We based our study on a corpus that's collected using an online version of the game *The Settlers of Catan*. Negotiations in this game mirror complex real life negotiations and provide a fruitful arena to study strategic conversation. Evaluation showed that our approach provides more accurate and complete information about trades than baselines that don't track how an offer evolves through the dialogue, and we also argued that game-theoretic reasoning about rational behaviour has advantages over relying on the presence or absence of an *Accept* act to make predictions.

Our approach, however, does not exploit discourse structure, which is needed to correctly handle long distance dependencies of offers on prior material. We will exploit this in future work to improve our results. We also plan to investigate other aspects of strategic reasoning on a larger dataset.

Conclusion

Main contributions

In this dissertation, we provided a method to extract preferences and model their evolution in spontaneous conversation. We assessed the reliability of the method on different corpus genres and applied it on a practical application by using it to predict trades in a win-lose game.

We started our presentation by an overview of the state of the art concerning preferences in the different domains of research our work relates: Game Theory, Artificial Intelligence and Natural Language Processing. We then proposed a study of preference change. We described several decision and game problems that illustrate how it is unavoidable to consider a dynamic model for reasoning about preferences and showed how CP-nets, a compact and qualitative representation of preferences, can be successfully used to handle it.

We studied how preferences are linguistically expressed in elementary discourse units on negotiation dialogues. We investigated preferences within dialogues with a common goal like fixing a meeting time (in our ***Verbmobil*** corpus) or making a reservation (in our ***Booking*** corpus) and in a competitive game, *The Settlers of Catan*, where the types of actions were more diverse (in our ***Settlers*** corpus). We proposed a new preference annotation scheme that requires two steps: (1) identifying the set of acceptable and non acceptable outcomes on which the agent's preferences are expressed, and (2) identifying the dependencies between these outcomes by using a set of specific operators expressing conjunctions, disjunctions and conditionals. The results showed that preference acquisition from linguistic actions is feasible for humans and that our scheme adapts relatively easily to different domains.

We presented an NLP-based approach to extract preferences. Similarly to the annotation process, we performed two steps: (1) extract the set of outcomes using machine learning techniques with a combination of local and discursive features, and (2) identify the preferences over the outcomes by using a hybrid approach combining both machine learning techniques and rule-based approaches. We assessed the reliability of our method on the ***Verbmobil*** and ***Booking*** corpora. For outcome extraction, our results showed that the dialogue discourse structure coupled with a top-level ontology are helpful to efficiently extract preferences. For preference identification, for each subtask, the results are in good agreement with the results obtained for the annotation by humans.

We proposed a method to model the evolution of the agents' preferences according to the dialogue moves. We extracted constraints on preferences and dependencies among them by exploiting discourse structure. Our method gave a formal description of each agent's preferences at any moment in the dialogue and modeled the evolution of these preferences as the dialogue progresses by constructing a dynamic partial CP-net description of preferences. The method supports qualitative and partial information about preferences, with CP-nets benefiting from linear algorithms for computing the optimal outcome from a set of preferences and their dependencies. The method relies on a study of 20 dialogues chosen at random from the ***Verbmobil*** corpus. We tested the algorithms predictions against the judgements of naive annotators and the results showed that our method is reliable.

We then combined the two previous works to predict trades in the win-lose game *The Settlers of Catan*. We estimated the preferences of EDUs automatically and exploited the conversation to dynamically construct a partial model of each players preferences in dialogues from our ***Settlers*** corpus, which in turn yielded equilibrium trading moves via principles from game theory. Specifically, we trained classifiers that map each utterance to its dialogue act (i.e., *Offer*, *Counteroffer*, *Accept*, etc.) and to other acts that are pertinent to bargaining (i.e., the resource types *Givable*, *Receivable*, etc.). And we developed a symbolic algorithm that, from the classifiers' output, dynamically constructs a CP-nets model of each player's preferences as the conversation proceeds (for instance, the preference to receive a certain resource, or to accept a certain trade). We adapt existing algorithms for computing equilibrium strategies in CP-nets to predict whether a trade is executed as a result of the players' negotiations, and if so predict who took part in the trade, and what they exchanged. We compared our model against four baselines and showed that our approach provides more accurate and complete information about trades than the baselines that don't track how an offer evolves through the dialogue.

Future work

Preferences and discourse structure

Currently, our automatic preference extraction tool relies on manually annotated discourse information following the Segmented Discourse Representation Theory

(SDRT). This is a first and necessary step before moving to real scenarios that rely on automatic annotations. The next step is to validate our results on automatically parsed data.

Our method for strategy prediction in the game *The Settlers of Catan* relies on a typology of dialogue acts that is domain sensitive. This approach, however, does not exploit discourse structure (since no automatic parser exists for our ***Settlers*** corpus), which is needed to properly handle long distance dependencies of offers. We also plan to exploit this in future work to improve our results and provide a domain independent method.

As far as we know, two SDRT-like parsers exist: the one developped for a dialogue corpus by Baldridge and Lascarides (2005b) and the one developped by our team (Muller et al., 2012). The first parser train a Probabilistic Context Free Grammar (PCFG) using dialogue-based features to produce tree representations of the discourse structure in dialogues from the Verbmobil corpus (Wahlster, 2000). Their best model achieves an F-score of 43.2% for labelled discourse relations and 67.9% for unlabelled ones. For labelled performance, the model is awarded a point for a span or relation which has the correct discourse relation label and both arguments are correct. For unlabelled, only the arguments need to be correct. The second parser performs a global A* search over the space of possible structures while optimizing a global criterion over the set of potential coherence relations. The system is evaluated on the Annodis corpus (Afantenos et al., 2012a), a collection of French discourse annotated newspaper and Wikipedia articles. It manages to achieve F-score of 46.8% for labeled reference structures and 66.2% for unlabelled ones.

Within the current European project STAC (ERC grant 269427), we plan to adapt Muller et al's parser (2012) to dialogues from our ***Settlers*** corpus and then to re-run our method to predict strategies with the discourse structure.

Preferences in medical data

While in this dissertation we focused on negotiation and bargaining dialogues (cooperative or not), in a work in progress we have started to study a completely different field, that is the medical domain. This work results from a collaboration with Diego Mollá and Abeed Sarker during a three-months stay at Macquarie University in Sydney. We give a quick overview of this work here and show how our study in negotiation dialogue is transposable in a really different corpus.

Our study concerns a corpus of medical questions (Mollá, 2010; Mollá and Santiago-Martínez, 2011).[4] The corpus contains entries from the *Journal of Fam-*

[4]See http://web.science.mq.edu.au/~diego/medicalnlp/

ily Practice[5] (JFP). Each entry concerns a question, e.g. *How can you prevent migraines during pregnancy?*, associated with an evidence-based answer. Even if this corpus was not built for the purpose of preference study (rather for text summarisation (Sarker et al., 2013)), several questions leads to discussion about finding the best option, e.g. *Which tests are the most useful for diagnosing PID?*, *What's the best treatment for pyogenic granuloma?*. Thus, the associated answers contain comparative information similar to what we found in our corpora. For example, we find sentences like *Cryotherapy is better than heat for treating acute muscle strain* where "*cryotherapy*" is a solution more recommended than "*heat*"; *SSRIs were equivalent to imipramine or alprazolam* where "*SSRIs*", "*imipramine*" and "*alprazolam*" are equally recommended. As the texts are written to be objective and give general answer to a problem, we can't find preferences in the sense that they are a subjective judgement by an agent. However the texts contains a lot of recommendation whose linguistic expression are similar to preferences and we find that our method could be pertinent to work in this kind of data.

For example, for the annotation, we can apply a two-step methodology as presented in Chapters 3 and 4: first, identify how the options are expressed, that is find the text spans that correspond to an option or not; then identify the ranking over the options, that is find which option is preferred (or equally preferred) to which other one.

The linguistic expressions are really similar to what we found in our corpora. Options are expressed within noun phrases which correspond to treatments or possible solutions for the question. We find the same kind of complex statements with disjunction (e.g., *The most cost-effective strategy is first to treat with azoles or undecenoic acid*) and conjunction (e.g., *Clarithromycin and erythromycin were similarly effective*). Recommendations can be expressed within comparatives and/or superlatives (e.g., *better than*, *as good as*, *the best treatment*). Expressions of opinions or preferences can also be used to indirectly introduce recommendations (e.g., *superior*, *useful*, *ineffective*, *prefer*). Recommendations can also be expressed via modalities (e.g., *Antibiotic treatment can eradicate bacterial vaginosis*, *Use of docusate sodium should be encouraged*).

Thus, for the automatic extraction, we can apply a method similar to what we presented in Chapter 4 where we can replace the domain ontology by the Unified Medical Language System[6] (UMLS) for example. Automatic outcome extraction achieves 60% for precision, 45% for recall and 51% for F-measure. While some errors will be hard to correct (for example when the classifier miss rare expressions), a lot of errors are due to annotation errors and we believe that these errors can be easily avoided by improving our annotation guide.

[5] http://www.jfponline.com/

[6] http://www.nlm.nih.gov/research/umls/

It would not be pertinent to apply our method for preference reasoning (from Chapters 5 and 6) to this corpus since it does not directly concerns the preferences of an agent. But still in the medical domain, we can envisage a lot of applications, like in patient-doctor interviews in which we could study how patients refine their preferences about different treatments.

From this quick overview, we perceive how our approach could be applied to several fields and for future work we want to apply and test the method in different applications from various domain in order to provide a general and domain-independent method for preference extraction and reasoning.

Towards Verbalised Preference elicitation

In a further perspective, we would like to apply the method to more practical applications like in a preference elicitation system in order to provide a tool where the users could express their preferences with Natural Language (NL). However, instead of translating NL user preferences into data base queries as in the *Expert-Clerk* system (Shimazu, 2001) (see Section 1.3.1), we could imagine a dialogue system (López-Cózar Delgado and Araki, 2005) that reasons on user's inputs and lets him refine his preference in NL.

Appendices

Appendix A

Glossary

A.1 The AGM model for belief revision 167
A.2 Evaluation measures for inter-annotator agreements 168
A.3 Evaluation measures for classification tasks 169
A.4 The McNemar's test 170

A.1 The AGM model for belief revision

We present the AGM model, named after its three originators Alchourrón, Gärdenfors and Makinson (1985). This presentation is a citation of Hansson's article (2006).

In the AGM framework, there are three types of belief change.

- In *contraction*, a specified sentence p is removed, i.e., a belief set K is superseded by another belief set K÷p that is a subset of K not containing p.
- In *expansion* a sentence p is added to K, and nothing is removed, i.e. K is replaced by a set K+p that is the smallest logically closed set that contains both K and p.
- In *revision* a sentence p is added to K, and at the same time other sentences are removed if this is needed to ensure that the resulting belief set K*p is consistent.

The two major tasks of a revision operator * are:

1. to add the new belief p to the belief set K, and
2. to ensure that the resulting belief set K*p is consistent (unless p is inconsistent).

The first task can be accomplished by expansion by p. The second can be accomplished by prior contraction by its negation ¬p. If a belief set does not imply ¬p, then p can be added to it without loss of consistency.

If a belief set does not imply ¬p, then p can be added to it without loss of consistency. This composition of suboperations gives rise to the following definition of a revision operator (the Levi identity): K*p = (K÷¬p)+p.

An operator * is an operator of partial meet revision if and only if it satisfies the following six postulates (where Cn(A) is the set of logical consequences of A):

- *Closure*: K*p = Cn(K*p).
- *Success*: p $\in$ K*p.
- *Inclusion*: K*p $\subseteq$ K+p.
- *Vacuity*: If ¬p $\notin$ K, then K*p = K+p.
- *Consistency*: K*p is consistent if p is consistent.
- *Extensionality*: If (p $\leftrightarrow$ q) $\in$ Cn($\emptyset$), then K*p = K*q.

These six postulates are commonly called the *basic Gärdenfors postulates for revision*.

A.2 Evaluation measures for inter-annotator agreements

For measuring agreement between our annotators, we use Cohen's kappa. The kappa is computed according to the following formula (Artstein and Poesio, 2008):

$$kappa = \frac{A_o - A_e}{1 - A_e}$$

- A_o is the observed agreement, that is, the proportion of items on which the two annotators agree. For a set of categories K, A_o is computed from $n_{a_i a_j k}$ the number of assignments to category k by both annotators a_i and a_j and i the number of items.

$$A_o = \frac{1}{i} \sum_{k \in K} n_{a_1 a_2 k}$$

- A_e is the agreement expected by chance. The chance of annotators a_1 and a_2 agreeing on any given category k is the product of the chance of each of them assigning an item to that category. For a set of categories K, A_e is computed from $n_{a_i k}$ the number of assignments to k by annotator a_i and i the number of items.

$$A_e = \frac{1}{i^2} \sum_{k \in K} n_{a_1 k} . n_{a_2 k}$$

A.3 Evaluation measures for classification tasks

The evaluative measures are computed from a confusion matrix like Table A.1 where:

- the True Positives (TP) are the number of predictions in class c that are correct.
- the False Positives (FP) are the number of predictions in class c that are incorrect.
- the False Negatives (FN) are the number of predictions incorrectly classified as non-c.
- the True Negatives (TN) are the number of predictions correctly classified as non-c.

		Actual class	
		c	non-c
Predicted class	c	TP	FP
	non-c	FN	TN

Table A.1: Confusion matrix.

The *Precision* is the proportion of predictions in class c that are correct.

$$Precision = \frac{TP}{TP + FP}$$

The *Recall* is the proportion of actual items in class c that are correctly predicted by the model.

$$Recall = \frac{TP}{TP + FN}$$

The *F-measure* (or *F-score*) is a combined measure that assesses the Precision/Recall tradeoff.

$$F\text{-}measure = 2.\frac{Precision.Recall}{Precision + Recall}$$

The *Accuracy* is the proportion of correct predictions (both Positives and Negatives).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

A.4 The McNemar's test

For determining whether one learning algorithm significantly outperforms another on a particular learning task, we use the McNemar's test based on a Chi-Square χ^2 distribution. For two learning algorithms A and B, we compute the following formula (Dietterich, 1998):

$$\frac{(|n_{MA\_OkB} - n_{OkA\_MB}| - 1)^2}{n_{MA\_OkB} + n_{OkA\_MB}}$$

- n_{OkA_MB} is the number of items misclassified by B but not by A,
- n_{MA_OkB} is the number of items misclassified by A but not by B.

If the resulting quantity is greater than $\chi^2_{1,0.95} = 3.841$ (the cut-off value for 95% significance level), then there is a significant difference between the two algorithms.

Annexe B

Résumé en français

Introduction **172**

B.1 Préférences : état de l'art **173**

B.1.1 Préférences en Théorie des jeux 174

B.1.2 Préférences en Intelligence Artificielle 174

B.1.3 Préférences dans le Traitement Automatique des Langues 177

B.2 Corpus et annotation des préférences **179**

B.2.1 Présentation des corpus 180

B.2.2 Comment les préférences sont-elles exprimées ? 184

B.2.3 Annotation des préférences 185

B.3 Extraction automatique des préférences **187**

B.3.1 Identification des options 188

B.3.2 Identification des préférences sur les options 192

B.4 Formaliser l'évolution des préférences **194**

B.4.1 Les règles d'évolution des préférences 194

B.4.2 Traitement d'un exemple 199

B.4.3 Conclusion 201

B.5 Prédiction de stratégies **202**

B.5.1 Pré-traitement : Prédiction des actes de dialogues et des préférences sur les ressources 202

B.5.2 Prédiction des actions stratégiques des joueurs 203

B.6 Conclusion **211**

Introduction

Dans la vie de tous les jours, qui n'a jamais été amené à exprimer ses préférences pour obtenir ce qui lui plait : à l'occasion d'un débat citoyen ou d'un achat, au cours d'un jeu contre un autre joueur, ou lors d'une prise de rendez-vous... Diverses méthodes et outils ont été proposés permettant aux utilisateurs d'exprimer leur(s) préférences(s) lors d'un processus de prise de décision. Par exemple, sur Internet, des sites proposent de trouver le vol le moins cher pour la destination et les horaires désirés ou le produit le plus adapté.

Traiter les préférences n'est pas aisé. Tout d'abord, il est nécessaire de connaître au moins partiellement l'ensemble des options sur lesquelles portent les préférences. Ensuite, il faut pouvoir définir un ordre a priori sur les options acceptables afin d'aboutir à un ensemble d'options optimales qui aidera l'utilisateur dans sa prise de décision. Cependant, définir cet ordre n'est pas trivial pour un utilisateur, surtout quand plusieurs critères entrent en jeu. Par exemple, on peut tenir compte de plusieurs critères pour choisir un nouvel appareil photo (tels que la durée de vie de la batterie, le poids, etc.). Ainsi, pour donner un ordre entre deux options (appareils photos), il faudra tenir compte des compromis et des interdépendances entre les différents critères. Ensuite, les utilisateurs manquent souvent d'informations complètes sur leurs préférences initiales qui tendent à changer au cours du temps. En effet, les utilisateurs peuvent apprendre du domaine, des préférences des autres et même de leurs propres préférences au cours du processus de prise de décision. Comme les agents ne connaissent pas complètement leurs préférences à l'avance, nous avons seulement deux moyens de les déterminer pour pouvoir raisonner ensuite : nous pouvons les inférer de ce que les agents disent ou de leurs actions non-linguistiques.

Plusieurs méthodes ont été proposées en Intelligence Artificielle pour éliciter les préférences (Chen and Pu, 2004) mais reposent souvent sur des interfaces spécifiques qui contraignent les utilisateurs. A notre connaissance, peu de travaux montrent comment les préférences pourraient être déterminées efficacement à partir de leur formulation en langue naturelle. Dans ce travail, nous analysons comment extraire et raisonner sur les préférences dans des conversations réelles de négociation et marchandage. C'est un travail à la croisée de la Théorie des Jeux, de l'Intelligence Artificielle et du Traitement Automatique des Langues.

Nous présentons d'abord, en Section B.1, l'état de l'art sur les préférences et présentons les définitions des termes importants pour la compréhension de ce travail. En Section B.2, nous présentons nos corpus de dialogues et étudions comment les préférences sont linguistiquement exprimées. Nous présentons ensuite la méthode d'extraction et de raisonnement sur ces préférences. Tout d'abord, nous détaillons en Section B.3 comment les préférences peuvent être extraites grâce à une combinaison d'apprentissage supervisé et de règles symboliques. En Section B.4, nous présentons des règles pour transformer les préférences extraites dans chaque tour de dialogue en une structure évolutive des préférences. Enfin, en Section B.5, nous combinons ces deux travaux (l'extraction automatique des préférences et la modélisation de leur évolution) pour prédire les échanges stratégiques des joueurs dans le jeu compétitif *Les Colons de Catane*.

B.1 Préférences : état de l'art

Tout d'abord, qu'est ce qu'une préférence ? Par préférence, on entend une notion de comparaison, donné par un agent, entre une entité et au moins une autre. Ces entités, appelées options, sont généralement comprise comme étant des actions que l'agent peut réaliser ou des états du monde qui sont le résultat direct des actions de l'agent. Par exemple, les préférences d'un agent peuvent porter sur l'action *acheter une nouvelle voiture plutôt que de garder l'ancienne* ou sur l'état du monde qui en résulte *avoir une nouvelle voiture plutôt que l'ancienne*. Les options sur lesquelles portent les préférences dépendent de la tâche ou du domaine étudié : elles peuvent concerner des objets à acheter, des restaurants où aller manger, des candidats à une élection, etc.

Parmi ces options, certaines sont acceptables pour un agent donné, c'est-à-dire que l'agent est prêt à les réaliser et d'autres sont inacceptables dans tous les cas. Parmi les options acceptables, l'agent en préfère généralement certaines par rapport aux autres. Notre méthode ne cherche pas à connaître l'option la plus préférée des agents mais à suivre l'évolution de leurs engagements sur ces options. Par exemple, si un agent propose de se rencontrer un certain jour à une certaine heure, alors on peut en déduire que parmi les options acceptables de l'agent il y en a une qui contient le fait de se rencontrer ce jour-là, à cette heure-là mais ce n'est peut être pas son option préférée. Il y a peut-être un autre moment qui lui conviendrait encore mieux.

Dans ce travail, on s'intéresse à l'étude linguistique des préférences, on cherche donc à modéliser les préférences telles qu'elles sont exprimées par les agents. On s'aperçoit que quasiment tout le temps, l'information donnée par les agents quand ils parlent correspond à une définition ordinale des préférences qui consiste à attribuer un ordre entre toutes les options envisageables, et pas à une définition cardinale qui permet de comparer les options en leur associant des valeurs numériques. L'ordonnancement des préférences peut être total (strict ou non), rendant chaque paire d'options comparable, ou partiel, quand certaines options ne peuvent pas être comparées par un agent donné. A cause de l'information imparfaite qu'ont les agents sur leurs préférences, le modèle de préférences qu'on utilise autorise de donner un ordre partiel.

Formellement, soit Ω l'ensemble des options possibles. Une *relation de préférence*, notée $\succeq$, est une relation binaire réflexive et transitive sur les éléments de Ω. Etant donné deux options o_1 et o_2, $o_1 \succeq o_2$ signifie que l'option o_1 est autant ou plus préférée que l'option o_2 pour l'agent qui exprime ses préférences. La relation de préférence stricte associée est $o_1 \succ o_2$ si et seulement si $o_1 \succeq o_2$ et $o_2 \not\succeq o_1$. La relation d'indifférence associée est $o_1 \sim o_2$ si $o_1 \succeq o_2$ et $o_2 \succeq o_1$.

B.1.1 Préférences en Théorie des jeux

Dans la théorie des jeux traditionnelle (Osborne and Rubinstein, 1994), les préférences ou les utilités sur les options sont déterminantes pour une prise de décision stratégique et rationnelle. Les options, dans la théorie des jeux standard, sont les états terminaux du jeux, les états finaux d'une stratégie complète, qui sont fonction de l'ensemble des joueurs $\mathcal{P}$ sur l'ensemble des actions $\mathcal{A}$. Les préférences sont souvent définies par une fonction d'utilité u qui associe des valeurs numériques à chaque option : $o_1 \succeq o_2$ si et seulement si $u(o_1) \geq u(o_2)$ et $o_1 \sim o_2$ et si et seulement si $u(o_1) = u(o_2)$.

Ce que nous appelons préférences est généralement capturé dans la théorie des jeux par une notion d'*utilité attendue* qui est définie en terme de *croyances* et de *fonction d'utilité*. Les croyances d'un agent sont représentées par une fonction de probabilité sur les états du monde. Et l'attrait d'un agent sur les conséquences d'une action sont représentées par la fonction d'utilité : les conséquences les plus attractives ont une valeur d'utilité plus grande. Pour l'ensemble des états du monde S, une *fonction de probabilité* p et une *fonction d'utilité* u, l'*utilité attendue* d'une action f est $\sum_{s \in S} p(s).u(f(s))$. Donc pour un agent rationnel, les actions préférées sont celles qui maximise sont utilité attendue et qui résultent donc d'un compromis optimal entre ce que l'agent préfère réaliser et ce que qu'il croit possible de réaliser.

B.1.2 Préférences en Intelligence Artificielle

Travailler avec les préférences implique trois tâches (Brafman and Domshlak, 2009; Kaci, 2011) : *acquérir* les préférences des utilisateurs, *modéliser* l'information liée à ces préférences et *raisonner* sur ces préférences (pour calculer la réponse à des requêtes communes telles que trouver l'option optimale, ordonner des ensemble d'options ou agréger les préférences de plusieurs utilisateurs pour prendre une décision collective).

B.1.2.1 Différents systèmes d'acquisition des préférences

Dans le domaine de l'Intelligence Artificielle, deux approches différentes sont utilisées pour extraire les préférences (Kaci, 2011; Pommeranz et al., 2012) : *l'apprentissage de préférences* où un système apprend les préférences à partir des préférences passées d'un agent afin de faire des prédictions sur ses préférences futures (Fürnkranz and Hüllermeier, 2011) et *l'élicitation de préférence* où les préférences sont le résultat d'un processus interactif avec l'utilisateur (Chen and Pu, 2004; Pu and Chen, 2008; Pommeranz et al., 2012).

Plusieurs systèmes exploitent différents types d'interactions avec les utilisateurs. Chen and Pu (2007) présentent un système de recommandation dans lequel les utilisateurs peuvent générer des critiques sur les caractéristiques des objets, par exemple pour un appareil numérique une critique multi-critères possible serait "Marque différente, Résolution plus faible et Moins cher". Les systèmes de recommandation *FindMe* (Burke, 2000) exploitent un raisonnement par similarité qui peut être filtré par des "tweaks" qui permettent de filtrer les options proposées par des contraintes telles que "moins cher" ou "plus grand" pour le système de recommandation d'appartements *RentMe*. Le système de recommandation *MovieLens* est un exemple de système qui exploite une interface permettant de noter les items grâce à des listes déroulantes (Miller et al., 2003). Dans le système *ExpertClerk* (Shimazu, 2001), un agent virtuel interagit avec un acheteur dans un langage naturel pour l'aider à trouver, comparer et choisir parmi un grand nombre de biens à acheter. Le système élicite les préférences du clients de deux façons : d'abord, en lui posant des questions, ensuite en lui proposant trois biens qui pourraient convenir et en observant sa réponse. Pour communiquer avec l'utilisateur, le système de dialogue utilisé est celui de Shimazu et al. (1992) qui fournit une interface en langue naturelle pour des bases de données. Il traduit les demandes de l'utilisateur en requêtes SQL. Le passage de l'un à l'autre est réalisé grâce à la reconnaissance de mots-clé et de motifs linguistiques associés au champs et aux valeurs de la base de donnée.

Les systèmes d'aide à la décision (Aloysius et al., 2006) élicitent les préférences pour aider les utilisateurs à prendre des décisions souvent critiques comme dans le domaine médical par exemple. Afin d'obtenir un modèle précis des préférences des utilisateurs, la majorité des systèmes d'aide à la décision représentent les préférences grâce à des fonctions d'utilités (cf Section B.1.1). Deux méthodes communément utilisées pour l'élicitation des préférences dans ces systèmes sont le jugement absolu (chaque attribut est indépendamment associé à un score sur une échelle absolue, par exemple pour une recherche d'emploi, on accorde une valeur de 8 sur 10 au salaire et une valeur de 6 à la distance lieu de travail/domicile) et la comparaison par paires (les attributs sont comparés deux à deux pour juger de leur importance relative, par exemple le salaire est plus important que la distance). Dans les interfaces d'élicitation associées, les scores sont généralement donnés par les utilisateurs en sélectionnant des items grâce à des listes déroulantes ou des radio-boutons pour des valeurs discrètes ou par des échelles à curseurs pour des valeurs continues (Pommeranz et al., 2012).

B.1.2.2 Les CP-nets, une méthode de formalisation des préférences

Différents formalismes existent pour représenter les préférences avec des approches diverses (graphiques, logiques, par contraintes, etc.). La plupart sont assez fortement liés et les préférences représentées dans un de ces formalismes peuvent être exprimées dans un autre par des équivalences ou de bonnes approximations (Coste-Marquis et al., 2004; Dubois et al., 2006; Domshlak et al., 2011). Parmi ces formalismes, nous avons choisi les CP-nets (Boutilier et al., 2004) pour modéliser les préférences de ce travail car ils présentent plusieurs avantages pour raisonner avec les préférences extraites de dialogues. En particulier, ils permettent de gérer des préférences partielles, qualitatives et conditionnelles telles qu'exprimées dans les conversations. Comme nous allons voir en Section B.5, ce choix est approprié puisqu'il nous permet de formaliser et raisonner avec succès sur les préférences représentées dans les CP-nets. Cependant, nous ne prétendons pas que ce soit le seul et meilleur formalisme pour modéliser les préférences telles qu'elles sont naturellement exprimées par les agents. Ce travail laisse la porte ouverte à une étude plus approfondie des pour et contre des différents formalismes dans un contexte conversationnel.

Les CP-nets, *conditional preference networks* en anglais, sont un langage graphique de formalisation des préférences basé sur la notion d'indépendance préférentielle conditionnelle. Ainsi, ils offrent une représentation compacte des préférences basée sur l'hypothèse *ceteris paribus*. Un CP-net $\mathcal{N}$ est $\langle \mathcal{G}, \mathcal{T} \rangle$ où

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe orienté où $\mathcal{V}$ est un ensemble de variables X_i associées à un domaine $\mathcal{D}(X_i)$ et représentées par des noeuds et $\mathcal{E}$ est un ensemble d'arc orientés entre les noeuds qui représentent les relations de dépendances préférentielles entre les variables. Dans le graphe, les prédécesseurs d'un noeud correspondant à la variable $X_i \in \mathcal{V}$ sont ses variables parents. Leur ensemble est noté $Pa(X_i)$. Ces variables influent sur les préférences de l'agent entre les différentes valeurs de X_i.

- $\mathcal{T}$ un ensemble de tables de préférences conditionnelles $CPT(X_i)$ associées à chaque $X_i \in \mathcal{V}$. Chaque table spécifie un ordre total sur les valeurs que peut prendre la variable X_i associée étant donné chaque combinaison possible des valeurs de ses parents. C'est-à-dire pour chaque instanciation p de $Pa(X_i)$, $CPT(X_i)$ spécifie soit $x_i \succ_p x_j$, soit $x_j \succ_p x_i$, soit $x_i \sim_p x_j$, $\forall x_i, x_j \in \mathcal{D}(X_i)$.

L'exemple suivant illustre ces définitions. Supposons un agent qui préfère aller de Paris à Hong-Kong par un vol de jour plutôt qu'un vol de nuit. S'il prend un vol de nuit, il préfère ne pas avoir d'escales mais s'il voyage de jour, il préfère en avoir pour rompre la monotonie du voyage. La figure B.1 présente le CP-net associé à cet exemple. La variable V correspond au moment du voyage. Son domaine est $\mathcal{D}(V) = \{v_j, v_n\}$ où v_j représente le voyage de jour et v_n de nuit. La variable E correspond aux escales. Son domaine est $\mathcal{D}(E) = \{e, \overline{e}\}$ où e représente un voyage avec escales et $\overline{e}$ sans.

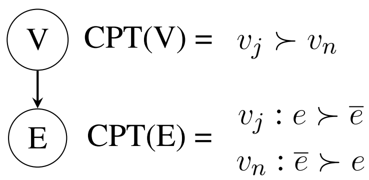


FIGURE B.1 – Un exemple de CP-net pour l'exemple du vol avec escale.

B.1.3 Préférences dans le Traitement Automatique des Langues

Dans la section précédente, nous avons présenté plusieurs systèmes pour l'élicitation des préférences. Même si ces systèmes aident les utilisateurs à trouver leurs préférences cachées et envisager différents critères pour lesquels des compromis sont nécessaires, dans certains cas ils peuvent être une contrainte plus qu'une aide, notamment quand l'interface n'est pas intuitive ou trop restrictive pour exprimer leurs préférences. Par exemple, qui n'a jamais été frustré devant une interface de réservation de trajets pour un voyage en train, avion ou autre. En général, de tels systèmes permettent de donner de l'information sur les préférences concernant les dates et la destination du voyage, parfois l'utilisateur peut également spécifier s'il préfère un vol direct et la catégorie des prix (économique, classe affaire ou première classe). Mais il est souvent impossible d'exprimer des contraintes plus spécifiques comme par exemple de vouloir chercher le vol le moins cher dans une fourchette de temps pour n'importe quelle destination d'un pays donné.

Pour ces problèmes pour lesquels l'utilisateur voudrait exprimer des contraintes spécifiques et plus généralement pour la majorité des problèmes de la vie réelle, nous pensons qu'il serait plus intuitif et efficace pour les utilisateurs d'exprimer leurs préférences dans un langage naturel sans la restriction de ce que le système autorise ou non. Ce genre d'approche résoudrait aussi les problèmes auxquels font face ces systèmes où la méthode doit fournir le bon compromis entre la complexité du système (qui doit être minimale pour que les utilisateurs puissent minimiser le temps et les efforts consacrés) et la performance (qui nécessite une connaissance la plus détaillée possible pour fournir des réponses précises). Exprimer les préférences dans le langage naturel éviterait aussi à l'utilisateur de fournir des efforts pour comprendre une interface spécifique et lui laisserait le contrôle de la précision de l'information qu'il est prêt à fournir. Un autre problème auxquels font face les systèmes d'élicitation existant est le choix des interfaces qui peut influencer les utilisateurs dans leur construction des préférences et nous pensons qu'une approche linguistique aiderait à réduire cette influence.

Cependant à notre connaissance, peu de travaux ont déjà étudié comment les préférences pourraient être efficacement extraites grâce à des méthodes du Traitement Automatique des Langues (TAL). Le système *ExpertClerk* (Shimazu, 2001) (cf Section B.1.2.1) est un premier pas pour une élicitation linguistique des préférences mais la méthode proposée dépend fortement de son association avec les bases de données. Des recherches en TAL propose une autre approche avec l'étude des opinions comparatives (Liu, 2010). Les opinions comparatives expriment des comparaisons entre deux objets ou plus basées sur les attributs qu'ils partagent (par exemple, "la qualité de l'image de l'appareil X est meilleure que celle de Y") (Jindal and Liu, 2006a; Jindal and Liu, 2006b; Ganapathibhotla and Liu, 2008). Mais ce travail est assez limité car soit il n'identifie la présence d'opinions comparatives qu'au sein d'une phrase sans extraire les informations sur les comparaisons, ou, quand il le fait, il ne considère que les comparaisons au sein de la phrase (pas de portée plus large) avec parfois même l'hypothèse qu'il n'y a qu'une seule relation de comparaison dans la phrase. Cependant, pour raisonner avec les préférences, il est incontournable de considérer des comparaisons plus complexes avec plus d'une dépendance à la fois et avec une plus large portée que la phrase.

De plus, il est important de ne pas ignorer que les préférences et les opinions sont des notions différentes. Alors que les opinions sont un point de vue, une croyance, un sentiment ou un jugement qu'un *agent peut avoir sur un objet ou une personne*, les préférences, comme nous les avons définies précédemment, impliquent un ordre de la part de l'agent et sont ainsi *relationnelles et comparatives*. Les opinions concernent donc un jugement absolu sur des objets ou des personnes (positif, négatif ou neutre), tandis que les préférences concernent un jugement relatif sur des options, les préférant, ou non, aux autres. Les exemples ci-dessous en sont une illustration :

(a) Le scénario de la première saison n'est pas mauvais.

(b) Le scénario de la première saison est meilleur que celui de la seconde.

(c) Je préfère revoir la première saison plutôt que la seconde.

(a) exprime une opinion directe positive sur le scénario mais nous ne savons pas si c'est le « plus » préféré. (b) exprime une opinion comparative entre deux saisons à propos de leurs caractéristiques communes (scénarios) et implique une préférence, ordonnant la première saison au-dessus de la seconde (Ganapathibhotla and Liu, 2008). Enfin, (c) exprime explicitement une préférence envers la première saison comparée à la seconde.

De plus, le raisonnement sur les préférences est différent du raisonnement sur les opinions. Les préférences d'un agent vont déterminer, s'il est rationnel, comment celui-ci va agir. Par contre, ce n'est pas le cas pour les opinions, ces dernières ont un lien beaucoup plus indirects avec les actions de l'agent. Par exemple, si je préfère acheter la voiture A plutôt que la voiture B et que je suis un agent rationnel, alors il est pertinent de prédire que je vais acheter la voiture B. Par contre, ce n'est pas parce que j'ai une bonne opinion de la voiture A ou B, qu'on pourra prédire que je vais l'acheter. Une opinion positive ou négative sur une chose, ne suffit pas à prédire qu'on va agir pour satisfaire cette chose. Les préférences mettent en jeu des mécanismes de décision beaucoup plus complets que les simples opinions.

B.2 Présentation des corpus et annotation des préférences

Nous proposons un nouveau schéma d'annotation pour étudier comment les préférences et les dépendances entre elles sont linguistiquement exprimées dans deux genres de corpus différents.

Nous présentons d'abord le schéma d'annotation proposé pour des dialogues entre deux agents coopératifs (les corpus ***Verbmobil*** et ***Booking***), puis nous montrerons comment nous avons étendu ce schéma pour annoter des jeux de négociation multi-joueurs entre agents non-coopératifs (le corpus ***Settlers***).

Nous décrivons la méthodologie adoptée pour l'annotation et détaillons ensuite les accords inter-annotateurs calculés sur chaque genre de corpus. Les résultats obtenus montrent que les préférences peuvent être correctement annotées par les humains et que le schéma d'annotation s'adapte assez facilement sur des domaines différents.

B.2.1 Présentation des corpus

B.2.1.1 Les corpus *Verbmobil* et *Booking* : des négociations coopératives

Nous utilisons deux corpus : un déjà existant, ***Verbmobil***, et l'autre que nous avons créé et appelé ***Booking***. ***Verbmobil*** a été utilisé pour créer la liste des traits d'apprentissage ; ***Booking*** a été utilisé pour évaluer à quel point notre méthode est dépendante du domaine.

Le premier corpus est composé de 19 dialogues choisis au hasard dans le corpus Verbmobil (Wahlster, 2000), dans lequel deux agents discutent pour fixer la date et le lieu d'un rendez-vous. En voici un exemple typique[1] :

(B.1)
π_1 A : Shall we meet sometime in the next week ?
Pouvons-nous nous rencontrer la semaine prochaine ?

π_2 A : What days are good for you ?
Quels jours te conviendraient ?

π_3 B : Well, I have some free time on almost every day except Fridays.
Eh bien, j'ai du temps libre quasiment tous les jours sauf les vendredis.

π_4 B : Fridays are bad.
Les vendredis sont vraiment mauvais.

π_5 B : In fact, I'm busy on Thursday too.
En fait, je suis également occupé jeudi.

π_6 A : Well next week I am out of town Tuesday, Wednesday and Thursday.
Eh bien, la semaine prochaine je ne suis pas là mardi, mercredi et jeudi.

π_7 A : So perhaps Monday ?
Donc peut-être lundi ?

Le second corpus a été construit à partir de plusieurs ressources d'apprentissage de l'anglais disponibles sur Internet[2]. Il contient 15 dialogues choisis au hasard, dans lesquels un agent, le client, appelle un service pour réserver une chambre, un vol d'avion, un taxi, etc. Dans ce corpus, la négociation porte sur le moment et les modalités de la réservation. En voici un exemple typique :

[1] Ces exemples sont présentés en anglais, leur langue d'origine, avec leur traduction en français. Dans la suite de ce résumé en français, nous ne présenterons plus que les traductions.

[2] par exemple, `http://www.bbc.co.uk/worldservice/learningenglish/`

(B.2) π_1 A : Northwind Airways, good morning. May I help you ?
Northwind Airways, bonjour. Puis-je vous aider ?

π_2 B : Yes, do you have any flights to Sydney next Tuesday ?
Oui, avez-vous un vol pour Sydney mardi prochain ?

π_3 A : Yes, there's a flight at 16 :45 and one at 18 :00.
Oui, il y a un vol à 16 :45 et un autre à 18 :00.

π_4 A : Economy, business class or first class ticket ?
Classe économique, affaire ou première classe ?

π_5 B : Economy, please.
Economique, s'il vous plaît.

Ces corpus sont associés avec une annotation de leur structure discursive d'après la Théorie des Représentations du Discours Structurées, SDRT (Asher and Lascarides, 2003). Dans cette théorie, chaque segment du discours est lié à un segment précédent par une ou des relations rhétoriques. Les segments sont soit des Unités de Discours Elementaires (UDE), c'est-à-dire qu'ils couvrent une seule clause du discours, soit des segments complexes composés d'autres segments. Les relations rhétoriques sont par exemple l'*Elaboration*, la *QAP* (paire de question-réponse), la *Correction*, la *Continuation* entre autres. Pour le corpus ***Verbmobil***, nous avons utilisé l'annotation de Baldridge and Lascarides (2005a). Pour le corpus ***Booking***, l'annotation a été faite par consensus en utilisant le même ensemble de relations rhétoriques qui a été utilisé pour annoter le corpus ***Verbmobil***. Pour illustrer l'annotation du discours, considérons l'exemple précédent de ***Verbmobil***. Les structures de discours correspondantes, pour les agents A et B, sont respectivement :

$Q\text{-}Elab(\pi_1, \pi_2) \wedge QAP(\pi_2, \pi) \wedge Plan\text{-}Elab(\pi_2, \pi) \wedge Plan\text{-}Elab(\pi_1, \pi_6) \wedge Plan\text{-}Elab(\pi_1, \pi_7) \wedge Plan\text{-}Elab(\pi_6, \pi_7)$,
et,
$Q\text{-}Elab(\pi_1, \pi_2) \wedge QAP(\pi_2, \pi) \wedge Plan\text{-}Elab(\pi_2, \pi)$
où : π : $Plan\text{-}Correction(\pi', \pi_5)$ et π' : $Explication(\pi_3, \pi_4)$.

Intuitivement, la question π_1 de A révèle sa préférence pour se rencontrer la semaine prochaine et *Q-Elab*(π_1, π_2) implique que n'importe quelle réponse à π_2 doit élaborer un plan pour réaliser la préférence révélée par π_1. Cela rend π_2 paraphrasable en « Quels jours te conviendraient la semaine prochaine ? », ce qui n'ajoute pas de nouvelles préférences. Cependant, la réponse de B dans les UDE π_3 à π_5 à la question élaborative π_2 de A révèle qu'il a adopté la préférence de A. En fait, la préférence de A est adoptée en π_3 qui spécifie une extension non-vide des jours où se rencontrer. Comme B donne une réponse étendue, les inférences sur ses préférences évoluent : de π_3 uniquement, on pourrait inférer une préférence pour une rencontre n'importe quel autre jour de la semaine prochaine que vendredi et son explication π_4 le confirmerait. Mais la correction π_5 oblige A à réviser ses inférences sur les préférences de B pour se rencontrer jeudi. Ces inférences à propos des préférences proviennent à la fois du contenu des énoncés de B et de la sémantique des relations qui les relient. La réponse π_6 de A révèle qu'il ne préfère pas mardi, mercredi et jeudi. Elle redéfinit ainsi les préférences révélées la dernière fois qu'il a parlé. La proposition suivante π_7 de A renforce l'inférence issue de π_6 que parmi lundi, mardi et mercredi (les jours que B préfère), A préfère lundi (même si ce n'était peut être pas son jour préféré quand le dialogue a commencé : peut-être que c'était vendredi). Le dialogue peut contraindre les agents à réviser leurs préférences lorsqu'ils apprennent des informations sur le domaine et sur leurs interlocuteurs.

Cet exemple montre que les préférences des agents dépendent de l'interprétation compositionnelle du discours sur les UDE. Les contraintes sont différentes pour différentes relations de discours, reflétant le fait que la sémantique des liens entre les segments a un impact sur la manière dont les préférences exprimées dans ces segments sont reliées entre elles.

Ces deux corpus sont relativement simple car ils impliquent seulement deux agents qui négocient pour satisfaire un objectif commun. Nous allons voir que le troisième corpus contient des négociations et marchandages plus complexes de par leur nature non-coopérative.

B.2.1.2 Le corpus *Settlers* : un jeu compétitif

Ce corpus concerne des dialogues du jeu *Les Colons de Catane*. C'est un jeu compétitif de type gagnant-perdant qui implique des négociations. Nous exploitons une version en ligne du jeu pour laquelle on enregistre l'état du jeu à tout moment et on l'aligne avec les conversations entre joueurs. Ces conversations impliquent du marchandage pour échanger des ressources mais aussi des commentaires sur des aspects stratégiques du jeu. Chaque joueur acquiert des ressources de cinq types (minerai, bois, blé, laine et argile) qu'ils utilisent dans différentes combinaisons pour construire des routes, des colonies et des villes qui leur rapportent des points de victoire. Les joueurs obtiennent les ressources à partir des lancers de dés ou à travers des échanges avec les autres joueurs.

Nous avons modifié la version en ligne déjà existante du jeu afin que les agents puissent puissent discuter les échanges à travers l'interface de chat. Nous avons ainsi recueilli une vingtaine de jeux pilotes impliquant le plus souvent des joueurs occasionnels. Chaque transcription de jeu contient une petite douzaine de dialogues de négociations (plusieurs segments de dialogues qui pris dans leur ensemble forme une négociation se suffisant à elle-même), pour un total d'environ 2000 tours de dialogues.

La plupart des tours de dialogues impliquent des négociations et représentent des offres, contre-offres, acceptation ou rejets d'offres et commentaires sur le jeu. Chacune de ces catégories, hormis la dernière, contient des préférences. Par exemple, l'énoncé « *Quelqu'un a-t-il de la laine en échange de blé ?* » implique plusieurs préférences. D'abord, il contient une préférence sur l'action de vouloir échanger avec quelqu'un d'indéterminé. Cet échange implique une préférence complexe pour recevoir de la laine plutôt qu'autre chose et étant donné qu'il recevra de la laine, l'agent préfère donner du blé plutôt qu'une autre ressource. Dans cet exemple, on ne sait pas ce que l'agent préfère s'il ne reçoit pas de laine. D'autres exemples sont encore moins spécifiés, comme par exemple « *Est-ce que quelqu'un a de la laine ?* ». Dans cet exemple, l'agent révèle la préférence de vouloir recevoir de la laine. Dans le jeu, ça implique qu'il sera prêt à offrir au moins une ressource en échange mais l'agent de révèlent aucune information concernant ses préférences sur les ressources qu'il est prêt à offrir. Ces exemples sont une illustration de plus du fait que les agents doivent raisonner avec des informations incomplètes.

(B.3) π_1 $Euan$: And I alt tab back from the tutorial. What's up ?
J'viens d'alt tab du tutorial. Quoi de neuf ?

π_2 $Joel$: do you want to trade ?
tu veux faire un échange ?

π_3 $Card.$: joel fancies a bit of your clay
joel voudrait un peu de ton argile

π_4 $Joel$: yes
oui

π_5 $Joel$: !

π_6 $Euan$: Whatcha got ?
T'as quoi ?

π_7 $Joel$: wheat
du blé

π_8 $Euan$: I can wheat for 1 clay or 1 wood.
J'peux du blé pour 1 argile ou 1 bois.

π_9 $Joel$: awesome
génial

Ce second dialogue de négociation est typique de ce qu'on peut trouver dans notre corpus. Il met en jeu un vocabulaire créatif (par exemple *alt tab* est utilisé comme verbe), des ellipses verbales (*J'peux du blé pour 1 argile*), des situations où la connaissance (ou mémoire) imparfaite des agents est évidente (le *Quoi de neuf ?*

de Euan). Il y a aussi des commentaires stratégiques, comme le segment π_3 et des tours de négociations sous-spécifiés (comme π_2 et π_7) qui sont complétés au fur et à mesure par de l'information qui donne de la connaissance commune.

B.2.2 Comment les préférences sont-elles exprimées ?

Notre objectif est d'analyser comment les préférences sont linguistiquement exprimées dans des segments de dialogues. Pour cela, deux étapes sont nécessaires : (i) identifier l'ensemble O des options sur lesquelles portent les préférences d'un agent, c'est à dire les termes, (ii) identifier les éventuelles dépendances entre les éléments de O en utilisant un ensemble d'opérateurs spécifiques, c'est à dire identifier les préférences de l'agent parmi les options énoncées. Pour illustrer ces étapes, prenons le segment suivant « Rencontrons nous lundi ou mardi ». Ici, nous avons deux options possibles, soit $O = \{lundi,\ mardi\}$. Ces options sont reliées linguistiquement par la conjonction « ou » qui signifie que l'agent est prêt à réaliser une de ces options, les préférant de manière égale. C'est cette dépendance qui nous permet d'inférer les préférences de l'agent et donc d'identifier, étant données deux options o_1 et $o_2 \in O$, la relation de préférence entre ces éléments (i.e. o_1 est préférée à o_2, o_2 est préférée à o_1 ou o_1 est indifférente à o_2.)

Dans une UDE, les préférences peuvent être exprimées de différentes manières. Elles peuvent être atomiques, par exemple, « Je veux X » ou « Je préfère X » où « X » est une option acceptable. Cette option peut être un nom comme « lundi », un groupe nominal comme « la semaine prochaine », un groupe prépositionnel comme « à mon bureau » ou un groupe verbal comme « se rencontrer ». Les préférences peuvent aussi être exprimées dans des constructions comparatives et/ou superlatives comme « une chambre moins chère » ou « le vol le moins cher ».

Les préférences sont aussi exprimées d'une manière indirecte en utilisant des questions. Bien que toutes les questions n'impliquent pas que l'auteur s'engage sur une préférence, dans beaucoup de cas elles le font. C'est-à-dire si un agent demande « Pouvons-nous nous rencontrer la semaine prochaine ? », il implique une préférence pour se rencontrer. Des expressions de sentiment ou de politesse peuvent aussi être utilisées pour introduire indirectement des préférences. Dans le corpus ***Booking***, le segment « Economique, s'il vous plaît » indique que l'agent préfère être dans la classe économique.

Chaque préférence atomique concerne une option qui est acceptable, ou non. Les options non acceptables sont exprimées via des négations qui indiquent ce que l'agent ne préfère pas. La négation peut être explicite, comme dans « Je ne veux pas qu'on se rencontre vendredi », ou inférée à partir du contexte, comme dans « Je suis occupé mardi ».

Les expressions de préférences peuvent aussi être complexes, exprimant des dépendances entre les options. En nous inspirant de la sémantique des CP-nets, *conditional preference networks* (Boutilier et al., 2004), nous reconnaissons que certaines préférences peuvent dépendre d'autres actions. Par exemple, étant donné que j'ai choisi de manger du poisson, je préfère boire du vin blanc plutôt que du vin rouge — ce que nous exprimons par $mangerPoisson : boireVinBlanc \succ boireVinRouge$.

Parmi les combinaisons possible de préférences, nous avons des disjonctions, conjonctions ou dépendances. Nous associons à chacune de ces expressions des opérateurs spécifiques (non-booléens) qui prennent en argument des préférences et que nous désignons respectivement par $\bigtriangledown$, $\&$ et $\rightarrow$. Nous présentons ci-après les descriptions linguistiques qui leur sont associées.

Les disjonctions de préférences expriment des choix libres. Par exemple dans « Rencontrons nous lundi ou mardi » ou dans « Je suis libre lundi et mardi », l'agent signifie que *lundi* et *mardi* sont des options acceptables et qu'il est prêt à réaliser une de ces options, en étant indifférent sur laquelle des deux choisir. Ainsi $o_1 \bigtriangledown o_2$ signifie $o_2 : o_1 \sim \overline{o_1}$, $\overline{o_2} : o_1 \succ \overline{o_1}$ et $o_1 : o_2 \sim \overline{o_2}$, $\overline{o_1} : o_2 \succ \overline{o_2}$.

Les conjonctions sont généralement exprimées par la coordination « et » comme dans « Pourrais-je avoir un petit déjeuner et un repas végétarien ? » ou bien « Lundi et mardi ne sont pas bons pour moi » où l'agent exprime deux préférences (respectivement sur les options acceptables *petit-déjeuner* et *repas végétarien*, et les options non-acceptables *pas lundi* et *pas mardi*) qu'il souhaite satisfaire et il aimerait en satisfaire au moins une des deux s'il ne peut pas les avoir toutes. Ainsi $o_1 \& o_2$ signifie $o_1 \succ \overline{o_1}$ et $o_2 \succ \overline{o_2}$.

Finalement, certaines UDE expriment des engagements sur des préférences dépendantes. Par exemple, dans la phrase « Pourquoi pas lundi, dans l'après-midi ? », il y a deux préférences : une pour le jour *lundi* et, étant donné la préférence pour *lundi*, une pour la période de l'*après-midi* (de lundi), au moins pour une des interprétations syntaxiques du segment. Ainsi, $o_1 \mapsto o_2$ signifie $o_1 \succ \overline{o_1}$ et $o_1 : o_2 \succ \overline{o_2}$.

B.2.3 Annotation des préférences

B.2.3.1 Annotation dans les corpus *Verbmobil* et *Booking*

Pour chaque UDE, deux annotateurs identifient comment les options sont exprimées et indiquent si l'option est acceptable, ou non, en utilisant l'opérateur unaire *not* et comment les préférences sur les options sont liées en utilisant les opérateurs binaires non-booléens &, $\bigtriangledown$ et $\mapsto$ qui prennent en arguments des expressions de préférences.

Alors que la forme logique d'un énoncé de préférence atomique est quelque chose de la forme $Pref(x)$, nous simplifions cela dans notre langage d'annotation, en utilisant uniquement l'expression de préférence x pour indiquer que l'agent préfère la valeur x par rapport aux autres alternatives, c'est-à-dire $x \succ \overline{x}$. Dans l'annotation de notre corpus ***Verbmobil***, x est typiquement un groupe nominal (GN) qui désigne un temps ou un lieu, x est ainsi un raccourci pour dire *se rencontrer à x*. Pour ***Booking***, x est un raccourci pour dire *réserver x*.

Nous proposons une annotation à deux niveaux : d'abord annoter les options, puis annoter leurs dépendances. Le premier niveau d'annotation servira pour évaluer notre méthode d'extraction des options (voir Section B.3.1) alors que le second niveau servira pour identifier les options préférées (voir Section B.3.2). Nous donnons ci-dessous un exemple de comment certains segments sont annotés. $< p > \_i$ indique que p est l'option numéro i dans le segment, et le symbole // est utilisé pour séparer les deux niveaux d'annotation.

π_1 : Je suis libre <à quatre>_1 ou <cinq heures>_2 <ces jours-là>_3. // 3 $\mapsto$ (1 *or* 2)

π_2 : <Mardi>_1, j'ai un séminaire <de 9h à midi>_2. // 1 $\mapsto$ *not* 2

B.2.3.2 Evolution du modèle d'annotation pour le corpus *Settlers*

Nous décrivons ici comment le schéma d'annotation précédent peut s'adapter pour annoter les préférences exprimées dans des conversations de marchandage sur une version en ligne du jeu *Les Colons de Catane* (corpus ***Settlers***). Le schéma d'annotation présenté dans la section précédente permet d'annoter des préférences dans des dialogues de négociation impliquant seulement deux agents ayant un objectif commun, fixer un rendez-vous pour ***Verbmobil*** ou arranger une réservation d'hôtel ou de vol d'avion pour ***Booking***. Pour annoter le corpus ***Settlers***, le schéma d'annotation est étendu afin de tenir compte d'un domaine plus complexe dans lequel les agents bien qu'ayant des intérêts stratégiques opposés sont amenés à former des coalitions pour échanger des ressources peu abondantes.

Dans ce corpus, les préférences sont plus complexes que pour ***Verbmobil*** et ***Booking***. Une option x peut concerner différentes actions. Dans « J'ai besoin de x », la préférence de l'agent est de recevoir la ressource x. Dans « J'offre x », la préférence de l'agent est de donner la ressource x. Dans « Je donne x pour y », l'agent exprime une préférence pour un échange complet qui implique de donner x et de recevoir y. Pour spécifier ces différentes actions dans l'annotation, nous utilisons en plus du vocabulaire déjà défini dans la section précédente, deux fonctions : *recevoir(p, d, <r,q>)* et *donner(p, d, <r,q>)* telles que : p est le propriétaire de la ressource, d est le destinataire, r est la ressource et q est la quantité de la ressource voulue (ou offerte). Si certains de ces arguments ne sont pas spécifiés, nous utilisons « ? ».

Pour chaque tour, les annotateurs identifient comment les options sont exprimées puis identifient les préférences grâce aux même opérateurs, not, &, $\bigtriangledown$ et $\mapsto$. Nous présentons ci-dessous un exemple.

π_1 Euan : J'ai <du blé>_1 pour <1 argile>_2 ou <1 bois>_3 // recevoir(Euan, Joel, <1, ?>) $\mapsto$ donner(Euan, Joel, <2,1> $\bigtriangledown$ <3,1>)

B.2.3.3 Evaluation du modèle

Pour évaluer notre modèle d'annotation sur les corpus, nous avons mesuré quatre taux d'accord inter-annotateurs concernant : (a) l'identification des options, (b) l'acceptabilité des options (opérateur not), (c) l'attachement des options et (d) l'identification des opérateurs de préférences (&, $\bigtriangledown$ et $\mapsto$). La table B.1 regroupe les résultats obtenus pour tous les corpus.

	Verbmobil	***Booking***	***Settlers***
identification des options (Kappa)	0,85	0,85	0,92
acceptabilité des options (Kappa)	0,90	0,95	0,97
attachement des options (F-mesure)	93%	82%	100%
identification des opérateurs (Kappa)	0,93	0,75	0,95

TABLE B.1 – Taux d'accord inter-annotateurs pour nos trois corpus.

B.3 Extraction automatique des préférences

Notre approche comporte deux étapes : (1) extraire les options exprimées dans chaque UDE. L'objectif est de repérer, au sein de chaque segment de dialogue, les expressions linguistiques sur lesquelles portent les préférences d'un agent ; (2) identifier les éventuelles dépendances entre les options extraites à l'étape 1 en utilisant un ensemble d'opérateurs spécifiques. C'est cette dépendance qui nous permet d'inférer les préférences de l'agent et donc d'identifier, étant données deux options, la relation de préférence entre elles.

Ce travail est réalisé sur les corpus ***Verbmobil*** et ***Booking*** présentés précédemment (voir section B.2.1). Cette méthode a été développée suite à l'étude de 20 dialogues du corpus ***Verbmobil***. Nous présentons les règles que nous avons élaborées et les illustrons sur un exemple de dialogue extrait de notre corpus d'étude.

B.3.1 Identification des options

Cette première tâche consiste à décider si un terme est une option de préférences ou non et donc de classer les termes en deux catégories : « *Option* » et « *Non option* » indiquant respectivement que le terme exprime une option faisant l'objet des préférences, ou non. Nous rappelons que les options peuvent être des noms, groupes nominaux, groupes prépositionnels ou groupes verbaux. Nous devons donc choisir quels groupes de mots doivent être classés.

Nous avons effectué ce travail sur deux des corpus présentés dans la section B.2.1, les corpus ***Verbmobil*** et ***Booking***. Dans les données, les agents négocient pour se mettre d'accord sur une action : se rencontrer un certain jour dans ***Verbmobil***, réserver un certain vol d'avion dans ***Booking***. Nous sommes généralement informés de ces actions dans les groupes verbaux. Cependant, les termes correspondants aux options de préférences sont plutôt contenus dans les groupes nominaux (GN). Par exemple, pour fixer un rendez-vous, la négociation porte sur les jours et les heures. Pour réserver un hôtel ou un vol d'avion, la négociation porte sur des options plus spécifiques comme « un vol direct », « une chambre double ». Il semble donc approprié d'extraire les GN. Pour les classer dans une des catégories « *Option* » ou « *Non option* », nous utilisons deux genres de traits : les traits locaux et les traits discursifs. Tous les traits sont binaires. Le classifieur est basé sur les Machines à Vecteurs de Support (SVM) (Burges, 1998). Un vecteur de traits est calculé pour chaque GN d'une UDE.

La portée des traits locaux est soit l'unité qui doit être classée, c'est-à-dire le GN, soit le segment qui contient le GN. Nous avons cinq traits au niveau du GN qui testent si le GN contient : (1) le label d'un concept appartenant à l'ontologie de domaine, (2) un comparatif, (3) un superlatif, (4) une disjonction ou (5) une conjonction. Nous avons dix traits au niveau du segment : (1) le voisin gauche du GN correspond à un label d'un concept de l'ontologie. Puisque la liste des termes associés à chaque concept de notre ontologie est courte, ce trait aide à retrouver des lexicalisations supplémentaires ; le segment contient (2) une disjonction ou (3) une conjonction ; le GN est dans la portée (4) d'une négation, (5) d'un modal ou (6) d'un verbe d'action du domaine (*se rencontrer*, *réserver*). La portée des négations et des modaux est résolue de manière simplifiée en utilisant l'arbre syntaxique de l'UDE ; le segment contient (7) un mot d'opinion (*bon*, *mauvais*, *OK*, etc.), (8) un mot de politesse ou (9) un mot qui introduit des préférences (*préférer*, *favori*, *choix*, *trop*, etc.) ; (10) le segment contient une référence à l'autre agent. Ce trait est un indice pour la classe « *Non option* ». Dans des segments comme *Tu as dit que tu n'es pas libre mardi matin ou mercredi après-midi ?*, l'agent n'apporte pas de nouvelle information sur les préférences mais répète seulement ce qui a déjà été établi par l'autre agent.

Nous avons neuf traits au niveau du discours : (1-6) les relations rhétoriques qui lient l'UDE courante à l'UDE précédente et à l'UDE suivante impliquent des préférences. Nous avons remarqué que certaines relations de discours peuvent aider à repérer des segments qui contiennent, ou non, des préférences. Nous dissocions les relations de discours en trois catégories : (a) celles qui impliquent « généralement » une « *Non option* » comme *Explication*, *Commentaire*, *Résumé*, (b) celles qui impliquent « peut-être » une « *Option* » comme *Elaboration*, *Continuation*, *Correction* et (c) celles qui impliquent « généralement » une « *Option* ». Dans ***Verbmobil***, 86 % des relations de discours sont de la catégorie (a) alors que 14 % des relations annotées appartiennent à la catégorie (b). Nous observons la même tendance pour ***Booking***. Il n'y a pas d'instances de la catégorie (c) dans les relations de discours utilisées lors de l'annotation des deux corpus. Ainsi, nous avons six traits : trois pour tester si la relation entre l'UDE courante et l'UDE précédente appartient, ou non, à une des trois catégories, et trois autre pour la relation entre l'UDE courante et l'UDE suivante ; (7-8) l'UDE courante ou l'UDE précédente est une question. Dans nos corpus, les formes interrogatives ne sont pas toujours suivies par une marque de question. Pour détecter les questions, nous utilisons donc les relations de discours spécifiques, comme *QAP*, *Q-Elab* ; (9) le GN apparaît au moins deux fois dans le dialogue.

Evaluation et Résultats. Plusieurs évaluations sont réalisées pour évaluer la validité de notre méthode d'extraction. La première est effectuée sur les 19 dialogues du corpus ***Verbmobil*** (C_V). Nous le séparons au hasard en un corpus d'entraînement constitué de 14 dialogues, soit 1337 GN, et un corpus de test de 5 dialogues, soit 347 GN. Dans la seconde (C_B), le classifieur est entraîné sur 11 dialogues du corpus ***Booking***, soit 623 GN et testé sur 4 dialogues pris au hasard, soit 214 GN. Pour la troisième, le classifieur est évalué en utilisant le corpus ***Verbmobil*** pour l'entraînement (les 19 dialogues) et le corpus ***Booking*** pour le test (les 15 dialogues) (C_V + C_B). Cette dernière évaluation, plutôt inhabituelle, est supposée aider à déterminer si notre méthode permet l'entrainement sur un corpus plus grand et disponible et le test sur un corpus plus petit et parfois d'un domaine différent. Pour toutes ces évaluations, nous utilisons le logiciel SVM-light [3].

Nous comparons les résultats du classifieur avec ceux de trois baselines : (1) la première classe tous les GN dans la catégorie « *Option* », (2) la seconde classe dans la catégorie « *Option* » tous les GN qui contiennent un concept appartenant à

[3] http ://svmlight.joachims.org

		C_V			C_B			$C_V + C_B$		
		P	R	F	P	R	F	P	R	F
Baselines	Tous les GN	40,9	100,0	58,1	28,0	100,0	43,8	28,3	100,0	44,1
	Ontologie seule	95,6	61,3	74,7	55,6	16,7	25,7	49,2	13,5	21,2
	Classifieur simplifié	65,2	71,1	68,0	68,4	43,3	53,1	43,9	55,7	49,1
Traits Locaux	Tous les traits (GN)	95,7	62,0	75,2	100,0	3,3	6,5	50,7	16,0	24,4
	+ Tous les traits (Segment)	94,1	78,9	85,8	68,4	43,3	53,1	60,2	26,2	36,5
Traits Discursifs	+ Relation Précédente	94,9	78,9	86,2	67,6	41,7	51,6	60,2	26,2	36,5
	+ Relation Suivante	94,0	77,5	84,9	66,7	40,0	50,0	59,4	25,3	35,5
	+ Questions	95,6	75,4	84,3	79,0	50,0	61,2	59,4	25,3	35,5
	+ $\geq$ 2 occurrences du GN	90,8	83,1	**86,8**	75,6	56,7	**64,8**	62,9	32,9	**43,2**

TABLE B.2 – Résultats (pourcentages) pour les trois évaluations.

l'ontologie, finalement (3) la troisième baseline est une version simplifiée de notre classifieur qui utilise seulement un sous-ensemble de nos traits (nous enlevons les traits basés sur l'ontologie ainsi que tous les traits basés sur les relations de discours).

La table B.2 présente les résultats, sous forme de précision (P), rappel (R) et F-mesure (F). Elle montre d'abord les résultats des baselines. Nous développons ensuite notre modèle en considérant les traits locaux au niveau du GN, puis nous ajoutons les traits locaux au niveau du segment et ajoutons progressivement les traits au niveau du discours (l'ajout des traits est symbolisé par le signe « + »). La dernière ligne présente le résultat final, obtenu en utilisant tous les traits.

Les résultats dans la table B.2 montrent que, parmi les trois baselines, la seconde donne les meilleurs résultats pour le corpus ***Verbmobil***. Ceci était attendu puisque l'ontologie a été construite pour ces données. Cependant, cette baseline ne permet pas de retrouver toutes les options car certains GN qui contiennent des concepts de l'ontologie ne sont pas des options (ce sont des répétitions, des commentaires, etc.) et bien sûr toutes les options exprimées par les agents ne sont pas « couvertes » par les concepts de l'ontologie. Pour le corpus ***Booking***, l'ontologie dégrade le rappel par rapport à la première baseline, puisqu'il y a un faible recouvrement entre les concepts dans l'ontologie et ceux dans le corpus. Il en va de même pour la troisième évaluation (C_V + C_B). Cependant, ce n'est pas un problème critique puisque des ontologies adaptées sont également disponibles pour le corpus de réservations (domaine du tourisme). Dans tous les cas, la troisième baseline donne des résultats assez stables, toujours meilleurs que ceux de la première baseline et, dans les deuxième et troisième évaluations (pour lesquelles nous n'avons pas utilisé d'ontologie adaptée), ces résultats sont également meilleurs que ceux de la deuxième baseline. Le classifieur donne un meilleur rappel pour la troisième évaluation que pour la deuxième. Cela peut montrer un problème de rareté des données lors de l'entrainement uniquement sur le corpus ***Booking*** (configuration (C_B)).

Les évaluations montrent que notre méthode d'extraction a une tendance similaire sur les corpus ***Verbmobil*** et ***Booking***. Nous voyons que les traits locaux au niveau du GN sont pertinents pour obtenir une bonne précision. Les traits au niveau du segment et les traits discursifs améliorent le rappel et la F-mesure dans les trois configurations. L'amélioration est mieux marquée dans les deuxième et troisième évaluations. C'est peut-être parce que l'ontologie, moins bien adaptée pour ces évaluations, a moins d'impact sur les performances. Finalement, pour le corpus ***Verbmobil***, nous obtenons une F-mesure de 86,8 %, i.e. presque 20 % au-dessus de la troisième baseline (classifieur simplifié) et plus de 10 % au-dessus de la deuxième baseline (basée sur l'ontologie). Pour le corpus ***Booking***, nous obtenons une F-mesure de 64,8 %, i.e. plus de 10 % au dessus du classifieur simplifié. Pour la troisième évaluation, les résultats ne montrent pas d'amélioration par rapport aux baselines. C'est probablement dû à l'influence de l'ontologie qui adapte mieux les vecteurs de support au corpus d'entrainement (***Verbmobil***), les rendant moins pertinents pour le corpus de test. En désactivant les deux traits basés sur l'ontologie, nous obtenons une précision de 50,2 %, un rappel de 62,9 % et une F-mesure de 55,8 %, soit une amélioration par rapport aux baselines.

Pour les traits discursifs, nous remarquons que, pour le corpus ***Verbmobil***, les relations rhétoriques entre l'UDE courante et le segment précédent apportent plus d'amélioration que les autres informations discursives. Cela peut s'expliquer par la nature du corpus, où le contexte (exprimé dans les tours de dialogues précédents) est important. Pour le corpus ***Booking***, le trait qui teste si l'UDE courante ou le segment précédent sont des questions apporte la meilleure amélioration des performances car ce corpus contient principalement des paires question-réponse. Pour la troisième évaluation, les traits discursifs n'apportent pas d'amélioration importante par rapport aux baselines. C'est peut-être causé par l'incapacité des informations discursives à compenser les différences entre les données d'entrainement et de test : en effet, en principe, il y a plus d'instances des traits locaux (au niveau du GN et du segment) associées à des cas positifs, que d'instances des traits discursifs associées à des cas positifs. Et quand le classifieur est entrainé sur des traits extraits d'un domaine de corpus et testé sur un autre domaine, le poids des traits discursifs peut ne pas suffire à compenser les autres traits, locaux.

Dans ces trois configurations, le trait testant la présence d'un GN au moins deux fois dans le dialogue apporte une amélioration conséquente par rapport aux autres traits. Cela était plutôt attendu puisqu'en principe la fréquence d'un GN apporte de l'information sur le sujet principal, et cela a du sens, puisque les agents ont tendance à exprimer des préférences sur le sujet de la discussion.

B.3.2 Identification des préférences sur les options

Une fois que l'ensemble des options O exprimées dans chaque UDE est extrait, l'étape suivante consiste à identifier comment ces options sont ordonnées. Pour cela, nous réalisons trois sous-tâches : (1) d'abord, pour chaque option, nous prédisons si elle est acceptable, ou non. Cela permet d'associer, ou non, à chaque option l'opérateur *not*. Par exemple, pour π_1 : « J'ai cours <mardi>_1 et <mercredi>_2 <de 9 h à midi>_3. », on obtient $1,\ 2,\ not\ 3$; (2) pour chaque UDE qui contient plus d'une option (environ 45% des UDE de notre corpus qui contiennent des options), nous construisons une représentation structurée des éléments de O afin d'obtenir des couples d'options. Par exemple, pour π_1 on obtient $((1,\ 2),\ not\ 3)$; (3) finalement, pour chaque couple d'options, on identifie récursivement l'opérateur qui lie les deux options. Par exemple, pour π_1 on obtient $((1,\ \bigtriangledown,\ 2),\ \mapsto,\ not\ 3)$. Cette *représentation des préférences de l'UDE finale* est ensuite traduite dans une représentation des CP-net en utilisant l'ensemble de règles spécifiques associées à chaque opérateur (voir section B.4).

Reconnaissance des options non-acceptables. Afin de reconnaître si une option est acceptable ou non, nous réalisons une tâche de classification binaire. Les options non-acceptables sont généralement dans la portée de négations comme *non, ne pas*, de mots avec connotation négative comme *mauvais* et d'expressions telles que *J'ai une réunion* qui ont une valeur de négation dans le contexte. Nous utilisons un ensemble de 9 traits : (1) l'UDE contient une négation ; (2) l'option est dans la portée de la négation d'après l'arbre syntaxique de l'UDE ; (3) il y a un marqueur discursif entre la négation et l'option ce qui en limite la portée ; (4) le nombre de mots de négation dans l'UDE ; (5) le nombre d'options dans l'UDE (6) les catégories syntaxiques des termes associés à l'option ; (7) les catégories syntaxiques des termes associés à la négation ; (8) le label de la négation et (9) le nombre de mots entre l'option et la négation la plus proche.

Nous réalisons une validation croisée avec 10 échantillons sur ***Verbmobil*** et ***Booking*** en utilisant le principe d'Entropie Maximale[4]. Nous obtenons une F-mesure de 89%. Les erreurs sont essentiellement dues à des problèmes dans la structure syntaxique ou à des négations implicites comme dans « <mardi>_1 j'ai une réunion <de 13 à 15h>_2 et une autre <de 16 à 18h>_3 » où l'option 3 est classée comme étant acceptable.

[4] http://nlp.stanford.edu/software/classifier.shtml

Reconnaissance des couples d'options. Pour réaliser cette sous-tâche, nous utilisons un ensemble de règles symboliques. Nous remarquons que dans la représentation structurée des options, la structure est semblable à celle du lien entre les noeuds équivalents dans l'arbre syntaxique. Dans π_1, les groupes nominaux « mardi » et « mercredi » sont des objets du verbe « avoir » et ont ainsi le même noeud père direct, tandis que le groupe nominal « de 9h à midi » appartient à un autre noeud de l'arbre. Nous obtenons ainsi la représentation structurée suivante $((1,\ 2),\ 3)$. Cependant, dans certains cas, l'ordre entre les options doit être renversé, principalement pour deux raisons : (1) la présence d'un marqueur de discours spécifique tel que « si » et « parce que », comme dans « <le 28>_1 je suis libre, <toute la journée>_2, si vous voulez faire <une réunion du dimanche>_3 », où nous avons $(3,\ (1,\ 2))$ puisque l'annotation complète associée est $3 \mapsto (1 \mapsto 2)$; (2) les options ne sont pas au même niveau ontologique, tels qu'un jour et une période de la journée, comme dans « ouais <l'après-midi>_1 c'est ok <pour mercredi>_2 » où nous avons $2 \mapsto 1$. On remarque également que dans quelques cas, certains marqueurs de discours introduisent un contraste (comme « mais », « bien que ») et introduisent une modification de l'ordre par rapport à l'arbre syntaxique, comme dans « J'ai cours <lundi>_1, mais, <n'importe quel moment après 13 ou 14h>_2 je suis libre » où nous avons $(1,\ (1,\ 2))$ puisque l'annotation associée est $not\ 1 \mapsto (1 \mapsto 2)$. Ce n'est pas facile de détecter les contrastes, en particulier quand les marqueurs discursifs sont ambigus comme « mais » qui introduit parfois du contraste (comme dans l'exemple précédent) et parfois non comme dans « j'ai une réunion qui commence <à 15h>_1, mais on peut se rencontrer <à 13h>_2 » où nous avons $not\ 1\ \&\ 2$. Les règles ont été construites par rapport au même ensemble de développement que pour l'extraction des options, c'est-à-dire 25 dialogues de ***Verbmobil*** et 21 dialogues de ***Booking*** qui contiennent 412 couples d'options. La F-mesure obtenue est de 81% pour ***Verbmobil*** et de 75% pour ***Booking***. Ces résultats sont en bon accord avec les résultats obtenus pour l'attachement des options lors de l'annotation (voir section B.2.3). Les erreurs proviennent à la fois du parseur (notamment pour l'attachement des coordinations) et des difficultés pour repérer les contrastes.

Reconnaissance des opérateurs de préférences. La dernière étape du processus est d'identifier comment les options de chaque couple défini à l'étape précédente sont liées en utilisant les opérateurs $\bigtriangledown$, & et $\mapsto$. Comme pour la sous-tâche précédente, nous utilisons un ensemble de règles définies à partir des 25 dialogues de ***Verbmobil*** et testées sut un ensemble de 31 dialogues provenant de ***Verbmobil*** et ***Booking***. Les résultats obtenus ***Verbmobil*** et ***Booking*** respectivement sont de (88%, 38%) pour $\&$, (96% , 71%) pour $\bigtriangledown$ et (96%, 69%) pour $\mapsto$. Ce qui donne une moyenne de 93% sur ***Verbmobil*** et 59% sur ***Booking***. Comme les humains, notre système échoue parfois à faire la différence entre & et $\mapsto$, entre $\bigtriangledown$ et $\mapsto$ et entre

& et $\bigtriangledown$. Les erreurs sont plus fréquentes pour le corpus ***Booking*** de par sa nature (des segments plus longs que dans ***Verbmobil*** qui rendent l'identification des dépendances entre options éloignées plus difficiles). Les erreurs dans ***Booking*** sont également dues à une correspondance moins claire entre les indices linguistiques et les opérateurs (voir notre discussion à la fin de la section B.2.3). Etant donné que les préférences impliquées par $o_1 \mapsto o_2$ et $o_1 \,\&\, o_2$ conduisent au même ensemble d'options préférées (l'agent préfère que o_1 et o_2 soient toutes les deux satisfaites), nous avons décidé de regrouper les opérateurs $\mapsto$ et & afin d'extraire, pour chaque UDE, les préférences sur la meilleure option. Ce qui conduit à une F-mesure moyenne de 98% pour ***Verbmobil*** et de 81% pour ***Booking***.

B.4 Formaliser l'évolution des préférences

Dans cette section, nous présentons une méthode pour modéliser l'évolution des préférences au cours du dialogue. Cette méthode se déroule en deux étapes : (1) d'abord, nous proposons un ensemble de règles utilisant les relations de discours pour intégrer les préférences de chaque UDE dans une description partielle des préférences en CP-nets (Boutilier et al., 1999) (cf Section B.4.1.1) ; (2) puis nous utilisons les contraintes liées au domaine et aux préférences pour obtenir le CP-net minimal et total associé de chaque agent (cf Section B.4.1.2). Nous illustrons l'application de la méthode sur un exemple dans la section B.4.2.

B.4.1 Les règles d'évolution des préférences

Au cours du discours, la connaissance sur les préférences exprimées par les agents évolue. Nous cherchons à modéliser cette évolution en utilisant les relations de discours pour mettre à jour une description partielle des CP-nets $\mathcal{DN}$. Pour cela, nous avons créé un ensemble de règles associées à chaque relation rhétorique pour définir leur influence sur la description $\mathcal{DN}$.

Pour modéliser l'influence du discours, nous utilisons la Théorie des Représentations du Discours Structurées, SDRT (voir Section B.2) pour obtenir la structure discursive du dialogue.

Pour modéliser l'évolution des préférences nous utilisons les CP-nets (voir Section B.1.2.2). Nous définissons un langage pour exprimer la description partielle des CP-nets. La formule $x_i \succ x_j(CPT(X_i))$ décrit un CP-net dans lequel une CPT contient une entrée de la forme $x_i \succ_p x_j$ pour une instanciation p de $Pa(X_i)$ qui peut être vide. La formule générale $\mathcal{N} \models y_1, ..., y_n : x_i \succ x_j(CPT(X_i))$ décrit un CP-net $\mathcal{N}$ dans lequel une CPT contient une entrée $x_i \succ_{\vec{u}} x_j$, également représentée par $\vec{u} : x_i \succ x_j$, où $y_1, ..., y_n$ apparaissent dans $\vec{u}$. De la même façon, nous définissons des formules sur la description des CP-nets $\mathcal{DN}$ qui permettent de décrire des contraintes sur les CP-nets des agents sans les spécifier complètement afin de respecter le fait qu'on n'a qu'une connaissance partielle de ces CP-nets. Nous verrons comment notre méthode permet de construire un CP-net minimal à partir d'une description partielle des CP-nets $\mathcal{DN}$ satisfiable.

Dans ces règles, la formule $\phi(P(\pi))$ signifie que ϕ est un élément décrivant $P(\pi)$, la description partielle des préférences sur lesquelles l'agent s'est engagé dans le segment π. X, Y et Z dénotent des variables sur lesquelles les préférences sont exprimées et $x, \overline{x}, y, \overline{y}, z, \overline{z}$ sont leurs valeurs associées. ϕ et ψ sont des formules représentant des préférences complexes et $Var(\phi)$ est la liste des variables dans ϕ. $Pa(X)$ est l'ensemble des variables parents de X et $\succ_X$ est la relation de préférences qui décrit la table conditionnelle $CPT(X)$.

Ces règles gèrent le type de préférences complexes rencontrées au cours du processus d'annotation (voir Section B.2). Nous rappelons la sémantique de ces opérateurs. *not* est utilisé pour représenter des préférences négatives, ainsi $not\ o_1$ signifie que o_1 est une option inacceptable. $\&$, $\bigtriangledown$ et $\mapsto$ représentent respectivement des conjonctions, disjonctions et conditionnelles. La description $\phi \& \psi(P(\pi))$ signifie que l'agent préfère que ϕ et ψ soient satisfaits et il préfère qu'au moins un des deux soit satisfait plutôt qu'aucun. $\phi \bigtriangledown \psi(P(\pi))$ signifie que l'agent préfère qu'au moins un de ϕ et ψ soit satisfait. $\phi \mapsto \psi(P(\pi))$ signifie que l'agent préfère que ϕ soit satisfait et si c'est le cas que ψ le soit aussi. Pour ϕ non satisfait, on ne connaît pas les préférences de l'agent sur ψ. Par exemple, pour le segment π : *Pourrions nous nous rencontrer lundi, dans l'après-midi ?*, la description des préférences associée est $lundi \mapsto aprem(P(\pi))$.

B.4.1.1 Des préférences dans chaque UDE à la description partielle des préférences

Commit introduit les préférences exprimées dans un segment de discours π dans la description des CP-nets $\mathcal{DN}$. La règle *Commit(π, $\mathcal{DN}$)* se décompose en plusieurs cas selon la forme de π. Nous présentons ci-dessous un des cas utilisé dans l'exemple de la section B.4.2.

Pour $x \mapsto y(P(\pi))$, nous obtenons :

- $\mathcal{DN} \models x \succ \overline{x}(CPT(X))$

- $X \in Pa(Y)$ et $\mathcal{DN} \models x : y \succ \overline{y}(CPT(Y))$[5].

IExplication, Elab, Plan-Elab et Q-Elab introduisent des dépendances entre préférences. Dans le cas de *IExplication* (π_1, π_2), les préférences exprimées dans π_2 expliquent celles de π_1, comme dans π_1 *: Je veux acheter un smartphone* π_2 *: pour ne plus me perdre.* Pour *Elab(*π_1, π_2*)*, la préférence dans π_1 est élaborée dans π_2, comme dans π_1 *: Je ne veux plus me perdre* π_2 *: donc je vais acheter un smartphone. Plan-Elab(*π_1,π_2*)* signifie que π_2 décrit un plan pour réaliser la préférence exprimée dans π_1. *Q-Elab(*π_1,π_2*)* est similaire à *Plan-Elab* mais le second constituant est une question.

1. *IExplication(*π_1, π_2*)*, *Elab(*π_2, π_1*)*, *Plan-Elab(*π_2, π_1*)* et *Q-Elab(*π_2, π_1*)* suivent la même règle.

   i Tout d'abord, la description $\mathcal{DN}$ est mise à jour en fonction de $P(\pi_2)$ en appliquant *Commit(*π_2, $\mathcal{DN}$*)*, si π_2 exprime une nouvelle préférence. Sinon, on peut directement passer à l'étape (ii).

   ii. Ensuite, la description $\mathcal{DN}$ est modifiée de manière à ce que chaque variable dans $P(\pi_1)$ soit dépendante des variables dans $P(\pi_2)$, i.e. $\forall X \in Var(P(\pi_1)), \forall Y \in Var(P(\pi_2)), Y \in Pa(X)$. Si π_1 exprime une préférence, la description $\mathcal{DN}$ est enrichie en fonction de $P(\pi_1)$, sinon, on ne fait rien.

L'étape (ii) dépend de la forme de π_1. Nous présentons ci-dessous le cas où $x \mapsto z(P(\pi_1))$ et $\phi(P(\pi_2))$ (l'agent exprime sa préférence sur $x \mapsto z$ par ϕ : il veut satisfaire x puis z si ϕ est satisfait). ϕ représente une formule de la description des préférences, ϕ' correspond à sa formule booléenne et $\overline{\phi'}$ à sa négation [6].

- Si $\succ_X$ n'est pas encore définie, on a $\mathcal{DN} \models \phi' : x \succ \overline{x}(CPT(X))$ et on ajoute $X \in Pa(Z)$. Sinon, il n'y a pas besoin de modifier $\succ_X$. En effet, c'est ce qu'on appelle une "*élaboration partielle*". Les variables qui ont été introduites depuis que la préférence sur X est apparue sont des parents de Z mais pas de X. Par exemple, si un agent s'engage sur $lundi$ puis sur $apresMidi$, et que plus loin dans le discours, il s'engage sur $lundi \mapsto a2heures$, alors $apresMidi$ est un parent de $a2heures$ mais pas de $lundi$.

[5] On remarquera que la description obtenue est la même que celle produite par la règle *Elab(*π_1, π_2*)* avec $X(P(\pi_1))$ et $Y(P(\pi_2))$ (voir règle 1). En fait, le symbole $\mapsto$ est un "raccourci" pour représenter une élaboration dont les deux arguments sont contenus dans le même segment de discours.

[6] Pour $\phi = y$, on a $\phi' = y$ et $\overline{\phi'} = \overline{y}$; pour $\phi = y\Delta z$ et $\phi = y \mapsto z$, on a $\phi' = y \wedge z$ et $\overline{\phi'} = \overline{y} \vee \overline{z}$; et pour $\phi = y \bigtriangledown z$ et $\phi = y\&z$, on a $\phi' = y \vee z$ et $\overline{\phi'} = \overline{y} \wedge \overline{z}$.

- Si $\succ_Z$ n'est pas encore définie, on a : $\mathcal{DN} \models \phi' \wedge x : z \succ \overline{z}(CPT(Z))$. Sinon, $\succ_{Z,(\phi' \wedge x)} = z \succ \overline{z}$, $\succ_{Z,(\phi' \wedge \overline{x})} = \succ_{Z,(\overline{\phi'} \wedge x)} = \succ_{z,(\overline{\phi'} \wedge \overline{x})} = \succ_Z$.

QAP étant une paire de question-réponse, son influence sur les préférences dépend du type de question.

2. *QAP*$_B$*(*π_1*,* π_2*)* et π_1 est une *question de type Oui/Non*

   - Quand π_2 est de type *Oui*, la description des préférences de B est mise à jour en appliquant *Commit(Elab*$_B$ *(*π_1*,*π_2*),* $\mathcal{DN}$*)* (ainsi la description des préférences de B inclut les préférences exprimées dans π_1 et π_2).
   - Quand π_2 est de type *Non*, si $P(\pi_1)$ et $P(\pi_2)$ sont consistants, alors la description des préférences de B est mise à jour par *Commit(Elab*$_B$*(*π_1*,* π_2*),* $\mathcal{DN}$*)*, sinon la description est mise à jour par *Commit(Correction*$_B$ *(*π_1*,*π_2*),* $\mathcal{DN}$*)* (voir la règle 4).

3. *QAP*$_B$*(*π_1*,* π_2*)* et π_1 est une *question ouverte*. Les préférences de B sur les variables dans π_1 et π_2 sont les mêmes que celles définies par la réponse *Oui* à une question de type Oui/Non. La description des préférences de B est mise à jour par *Commit(Elab*$_B$*(*π_1*,* π_2*),* $\mathcal{DN}$*).*

Correction et Plan-Correction permettent aux agents de rectifier leur engagement sur les préférences. Elles peuvent avoir plusieurs effets sur les préférences. Premièrement, elles peuvent corriger une entrée, i.e. étant donné *Correction(*π_1*,* π_2*)*, certaines variables de $P(\pi_1)$ sont remplacées par des variables de $P(\pi_2)$. Par exemple dans π_1 *: On peut se voir Jeudi.* π_2 *: Non, désolé, j'ai un séminaire toute la journée*, chaque occurrence de $jeudi$ est remplacée par $\overline{jeudi}$ et vice versa).

4. *Correction(*π_1*,* π_2*)* et $X \in Var(P(\pi_1))$ est remplacé par $\{Y_1, ..., Y_m\} \in Var(P(\pi_2))$. Si $Pa(X) = \emptyset$, on ajoute $\mathcal{DN} \models y_k \succ \overline{y}_k(CPT(Y_k))$ pour tout $k \in \{1, \ldots, m\}$ et on enlève $x \succ \overline{x}(CPT(X))$ (ou $\overline{x} \succ x(CPT(X))$). Sinon, on remplace chaque description de $CPT(X)$ par une description de $CPT(Y_k)$ équivalente (où x est remplacé par y_k pour tout $k \in \{1, \ldots m\}$).

Deuxièmement, *Plan-Correction* peut aussi aboutir à la correction du propre plan de l'agent suite aux propositions de l'autre agent. Par exemple, dans un dialogue, tous les agents s'accordent sur le fait de se rencontrer la semaine prochaine, ainsi dans la description de leurs CP-nets, il y a l'entrée $semn1 \succ \overline{semn1}$. Par la suite, la discussion montre que les disponibilités de chacun ne sont pas compatibles et l'un dit *"Bon, on ne pourra pas se rencontrer cette semaine là."*. Cela ne signifie pas que l'agent préfère $\overline{semn1}$ à $semn1$. C'est pourquoi, il ne faut pas corriger la description des CP-nets en utilisant la règle précédente. On crée une nouvelle règle qui corrige ce qu'on appelle la *liste opérative*, c'est-à-dire la liste de variables dont dépendent les futures préférences. Dans l'exemple précédent, on enlève $semn1$ de la liste opérative.

5. Pour *Plan-Correction(*π_1, π_2*)* qui corrige la liste des variables parents, la liste opérative devient l'intersection de tous les $Pa(X)$ où $X \in Var(P(\pi_1))$. On applique ensuite *Commit(Plan-Elab(*π_1,π_2*),*$\mathcal{DN}$*)*, si $P(\pi_2)$ contient une nouvelle préférence ϕ. Si le CPT affecté par cette élaboration ne contient pas d'entrée pour la liste opérative courante $\mathcal{O}$, alors l'entré $\mathcal{O}\colon\ \phi$ doit être ajoutée à la description $\mathcal{DN}$.

Continuation, Contraste, et Q-Cont suivent les règles associées à la relation continuée (par rapport à la hiérarchie du dialogue). Plusieurs autres règles, **Explication**, **Resultat**, **Qclar** (question de clarification), **Commentaire**, **Résumé**, et **Confirmation** ont soit aucun effet, soit les mêmes effets qu'une *Elaboration*. Nous ne les détaillons donc pas plus.
Parfois, au cours du processus d'application des règles, la description des CP-nets $\mathcal{DN}$ obtenue peut être insatisfiable (voir section suivante). Si une évaluation du CP-net montre que la description n'est pas satisfiable alors on applique la règle 4, associée à la *Correction*.

B.4.1.2 De la description des préférences au CP-net minimal associé

Les règles définies dans la section précédente sont indépendantes du domaine et permettent de compléter de façon monotone la description des CP-nets $\mathcal{DN}$. Cependant, ces règles ne permettent pas de trouver une description totale des CP-nets, c'est-à-dire une description qui associe à chaque table de préférences conditionnelles $CPT(X)$ des entrées pour toutes les combinaisons possibles des valeurs des parents de X. C'est pourquoi, nous avons besoin d'une nouvelle étape qui complète la descriptions grâce aux contraintes du domaines et aux propriétés des préférences (transitivité, irréflexivité).
Dans le cas du domaine du temps sur lequel porte notre corpus de travail, il est important de noter que des variables telles que $Lundi$ signifie "se rencontrer lundi", et $lun \mapsto rdv$. Ainsi, lorsqu'on a une description incomplète des CP-nets avec $rdv\ :\ lun \succ \overline{lun}$ mais aucune information dans le cas où on a $\overline{rdv}$, d'après la remarque précédente on peut en déduire le cas manquant $\overline{rdv}\ :\ \overline{lun} \succ lun$. De plus, le domaine apporte des informations concernant d'éventuelles inconsistances : on sait par exemple que le matin et l'après-midi sont distincts. On ne peut donc pas avoir d'entrée comme $mat \succ \overline{mat}$ et $mat\ :\ aprem \succ \overline{aprem}$.

Grâce aux connaissances du domaine, on peut également traiter les cas où certaines préférences s'expriment sur des dates, par exemple *le 4*, et d'autres sur des jours déterminé par rapport au contexte, par exemple *jeudi prochain*. De plus, on peut simplifier le nombre de variables utilisées. Par exemple, dans la description $rdv \wedge \overline{lun} \wedge \overline{mar} \wedge mer : aprem \succ \overline{aprem}$, on peut reconnaître les valeurs qui se rapportent aux mêmes concepts, par exemple les jours, et obtenir la description simplifiée $rdv \wedge mer : aprem \succ \overline{aprem}$.
Lors de cette étape, nous cherchons à tester la satisfiabilité de la description des CP-nets $\mathcal{DN}$. Pour cela, nous vérifions que les formules obtenues sont conformes à ces nouvelles contraintes et nous transformons ces formules en des entrées des CP-nets. Afin que les CP-nets soient complets, nous vérifions pour chaque variable X que le CPT associé possède une entrée pour toutes les combinaisons de valeurs et si ce n'est pas le cas, nous ajoutons l'entrée $x \sim \overline{x}$ pour les combinaisons manquantes. Si la description des CP-nets respecte toutes ces spécifications, alors nous obtenons le CP-net minimal. Il est important de noter que ces tests de satisfiabilité peuvent être réalisés à n'importe quel tour du processus, ce qui est très utile pour repérer les inconsistances qui peuvent apparaître au cours du dialogue et qui nécessitent d'appliquer la règle de la correction.

B.4.2 Traitement d'un exemple

Pour illustrer le fonctionnement de nos règles, nous présentons leur application sur un exemple traduit d'un extrait de dialogue de notre corpus de travail. La figure B.2 présente la structure discursive associée à ce dialogue.

- π_1 contient un *engagement* sur la préférence de se rencontrer que l'on représente par la variable RDV qui peut prendre les valeurs rdv et $\overline{rdv}$. Cette préférence est élaborée par l'engagement sur le fait de se rencontrer la journée du 27. On la représente par la variable $J27$ qui peut prendre les valeurs 27 et $\overline{27}$. On a donc $rdv \mapsto 27(P(\pi_1))$ et on applique *Commit*$_A$(π_1, $\mathcal{DN}$) qui donne :
  $\mathcal{DN}_A \models rdv \succ \overline{rdv}(CPT(RDV))$, et
  $\mathcal{DN}_A \models rdv : 27 \succ \overline{27}(CPT(J27))$.

- π_2 est la *réponse de type Non à la question Oui/Non* exprimée dans π_1. Comme les préférences dans π_1 et celles dans π_2 ne sont pas consistantes, on applique *Commit(Correction*$_B$(π_1,π_2), $\mathcal{DN}$) (voir règles 2 et 4). On a $\overline{27}(P(\pi_2))$ et d'après la règle de la correction, l'agent B récupère les engagements de A mais remplace toutes les occurrences de 27 par $\overline{27}$ et vice versa. On obtient la description suivante :
  $\mathcal{DN}_B \models rdv \succ \overline{rdv}(CPT(RDV))$, et
  $\mathcal{DN}_B \models rdv : \overline{27} \succ 27(CPT(J27))$.

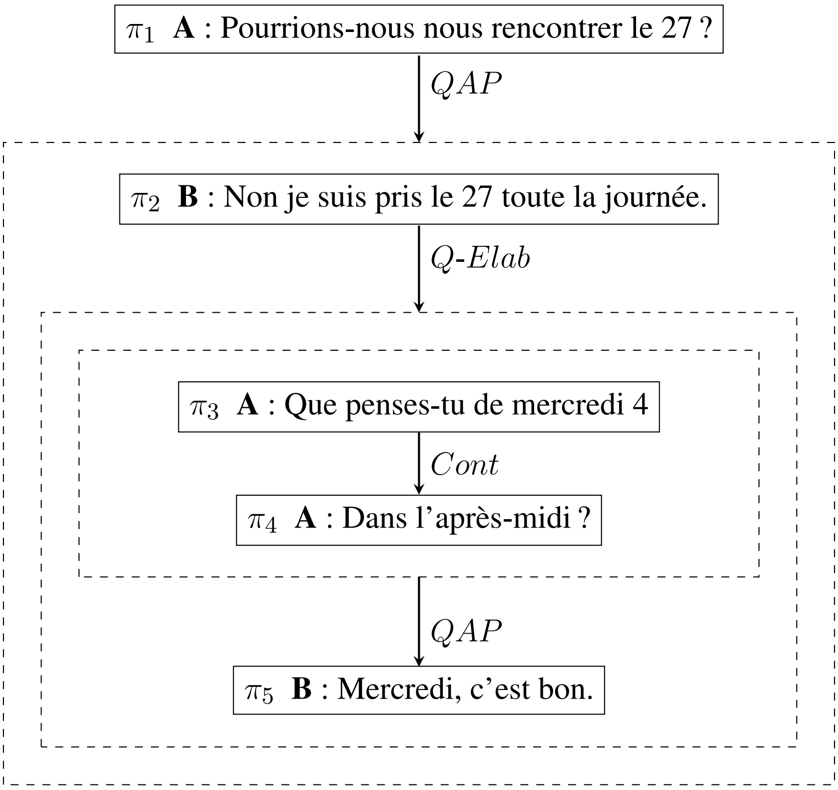


FIGURE B.2 – Structure discursive d'un dialogue de ***Verbmobil***.

π_3 est lié à π_2 par une *question élaborative*. On applique *Commit*$_A$(*Q-Elab*$_A$(π_2, π_3) $\mathcal{DN}$) (voir règle 1) avec $mer4(P(\pi_3))$. On obtient la description suivante :
$\mathcal{DN}_A$ reste identique pour $(CPT(J27))$. En effet, l'agent A met à jour ses plans concernant le 27 par rapport aux informations données par l'agent B mais il ne modifie pas ses préférences. Par contre, il met à jour la liste des variables opératives qui ne contiendra plus 27 mais $\overline{27}$ (voir règle 5 pour la Correction).
$\mathcal{DN}_A \models rdv \wedge \overline{27} : mer4 \succ \overline{mer4}(CPT(J4))$. Grâce à la connaissance du domaine, on sait que les variables $J27$ et $J4$ concernent le fait de se rencontrer un certain jour donc 27 et $mer4$ sont incompatibles. Ainsi, on peut compléter la description par $\mathcal{DN}_A \models rdv \wedge 27 : \overline{mer4} \succ mer4(CPT(J4))$ (voir section B.4.1.2).

π_4 est lié par une *Continuation* à π_3 donc c'est toujours la règle de *Q-Elab* qui s'applique. On applique donc *Commit*$_A$(*Q-Elab*$_A$(π_3, π_4) $\mathcal{DN}$) (voir règle 1).

On a $am(P(\pi_4))$ où am représente l'après-midi et on obtient :
$\mathcal{DN}_A \models rdv \wedge \overline{27} \wedge mer4 : am \succ \overline{am}(CPT(Am))$ que l'on peut simplifier en $\mathcal{DN}_A \models rdv \wedge mer4 : am \succ \overline{am}(CPT(Am))$ grâce à la connaissance du domaine (voir section B.4.1.2). Comme on n'a pas d'information sur la préférence de se rencontrer l'après-midi dans le cas où on se rencontre un autre jour que le 4, on complète la description par $\mathcal{DN}_A \models rdv \wedge \overline{mer4} : am \sim \overline{am}(CPT(Am))$.

π_5 est la *réponse à la question ouverte* posée en $[\pi_3 - \pi_4]$. D'après la règle 3 associée à QAP, on applique donc *Commit*$_B$(*Elab*$_B$(π_4, π_5) $\mathcal{DN}$). On a $mer4(P(\pi_5))$ et on obtient la description suivante :
$\mathcal{DN}_B \models rdv \wedge \overline{27} : mer4 \succ \overline{mer4}(CPT(J4))$, $\mathcal{DN}_B \models rdv \wedge 27 : \overline{mer4} \succ mer4(CPT(J4))$, et
$\mathcal{DN}_B \models rdv \wedge mer4 : am \succ \overline{am}(CPT(Am))$ et $\mathcal{DN}_B \models rdv \wedge \overline{mer4} : am \sim \overline{am}(CPT(Am))$.

Comme la variable $J27$ correspond au fait de se rencontrer la journée du 27, on complète les descriptions par $\mathcal{DN}_A, \mathcal{DN}_B \models \overline{rdv} : \overline{27} \succ 27(CPT(J27))$ (voir section B.4.1.2). De la même façon, on complète les descriptions de $CPT(J4)$ et $CPT(Am)$.

B.4.3 Conclusion

Nous avons proposé une méthode d'élicitation des préférences qui applique un algorithme indépendant du domaine pour construire une description partielles des préférences. Celui-ci est suivi par une méthode spécifique au domaine qui identifie le CP-net minimal satisfaisant la description partielle ainsi que les contraintes liées au domaine. Ainsi, la méthode est capable de gérer les connaissances partielles des préférences qui découlent des dialogues ainsi que leur évolution au cours des échanges entre les agents.
Nous avons évalué ce travail en comparant nos prédictions à celles de trois annotateurs naïfs sur trois textes du corpus que nous n'avons pas utilisé pour construire nos règles. Pour cela, nous leur avons demandé de dire pour chaque UDE si elle introduit une préférence ou met à jour, corrige ou supprime une préférence déjà introduite dans une précédente UDE. Pour la relation d'*Elaboration*, la moyenne de l'accord juge-système est de 91% et l'accord entre juges est de 92,7%. Puisque cette relation est utilisée dans les règles d'autres relations, ce résultat évalue les règles 1, 2 (*Oui*) et 3. Pour la relation de *Correction*, la moyenne de l'accord juge-système est de 85,7% et l'accord entre juges est de 81%. Ce résultat évalue les règles 2 (*Non*), 4 et 5.

B.5 Prédiction de stratégies

Dans cette section, nous proposons de combiner les méthodes des deux précédentes sections (l'extraction automatique des préférences vue en Section B.3 et la modélisation de l'évolution des préférences vue en Section B.4) pour prédire les actions des agents dans le jeu stratégique *Les Colons de Catane*. Nous exploitons les tours de conversation pour construire dynamiquement un modèle partiel des préférences de chaque joueur, qui à son tour permet de prédire si oui ou non, un échange est exécuté en résultat de la négociation entre joueurs, et si c'est le cas, nous prédisons qui a pris part dans l'échange et ce qu'ils ont échangé. Notre méthode montre des résultats prometteurs, meilleurs que ceux des baselines qui ne capturent pas correctement l'évolution des préférences.

La méthode repose sur trois niveaux : (1) une caractérisation de chaque tour de dialogue en terme d'actes de dialogue qui sont spécifiques aux jeux de marchandage (*Offres*, *Contre-offres*, etc.), (2) l'identification des préférences des joueurs (par exemple, une préférence pour recevoir une certaine ressource, la préférences pour accepter de faire un échange, etc.) et finalement (3) nous combinons les deux précédents niveaux en une description partielle des préférences de chaque agent grâce aux CP-nets (présentés en Section B.1.2.2) que nous utilisons pour prédire le comportement stratégique des joueurs.

Notre travail utilise le corpus ***Settlers*** présenté en Section B.2.1.2. Les deux premiers niveaux sont obtenus par apprentissage supervisé. Ensuite, à partir de la sortie des classifieurs, nous développons un algorithme symbolique qui permet de construire les CP-nets du troisième niveau et de prédire les échanges qui sont exécutés pendant le jeu.

B.5.1 Pré-traitement : Prédiction des actes de dialogues et des préférences sur les ressources

B.5.1.1 Identification des des actes de dialogues

Le corpus ***Settlers*** a été annoté avec 5 types d'actes de dialogues en lien avec la tâche de négociation : *Offre*, *Contre-Offre* (qui peut-être la réponse à une offre ou la spécification d'une offre déjà proposée par le joueur), *Acceptation* ou *Refus* (d'une offre adressée au joueur) et *Autre* qui regroupe les tours de dialogues non-pertinents pour la négociation tels que des commentaires sur le fonctionnement du jeu.

Comme les actes de dialogues sont dépendants les uns des autres (par exemple une *Acceptation* ou un *Refus* suit généralement une *Offre* ou une *Contre-offre*), nous utilisons les champs aléatoires conditionnels, Conditional Random Fields en anglais (CRF) pour apprendre ces actes de dialogues. Nous utilisons un ensemble de traits lexicaux, syntactiques et sémantiques que nous combinons en unigrammes et bigrammes. Nos données contiennent 511 dialogues découpés en 2741 UDE dont 410 sont associées à la catégorie *Offre*, 197 à *Contre-offre*, 179 à *Acceptation*, 398 à *Refus* et 1557 à *Autre*. Pour évaluer cette prédiction, nous utilisons la validation croisée sur 10 échantillons et nous présentons les résultats dans la table B.3.

Acte de dialogue	Précision	Rappel	F-mesure
Autre	87,4	93,1	90,1
Offre	80,0	81,0	80,5
Contre-of.	64,8	53,3	58,5
Acceptation	65,1	53,1	58,5
Refus	81,7	73,9	77,6
F-mesure moyenne			**73,0**
Accuracy			**83,0**

TABLE B.3 – Résultats pour la classification des actes de dialogues.

B.5.1.2 Identification des préférences sur les ressources

Comme le vocabulaire des ressources du jeu *Les Colons de Catane* est un ensemble fermé de mots (*bois*, *argile*, etc.) et de leurs synonymes, nous reconnaissons les ressources simplement par la présence ou non d'un mot du lexique dans les groupes nominaux (GN). Notre corpus contient 4361 GN dont 21% sont des ressources et 79% n'en sont pas. Cette méthode basique obtient une F-mesure de 96,9%.

Le corpus a été annoté avec quatre type d'information sur les ressources : *Donnable, Recevable, Non-Donnable* et *Non-Recevable*. Nous prédisons ces types en utilisant les CRFs avec des unigrammes et bigrammes provenant de traits locaux à l'UDE. Notre corpus contient 1077 Ressources dont 510 *Recevable*, 432 *Donnable*, 116 *Non-Donnable* et 19 *Non-Recevable*. Les résultats sont obtenus par validation croisée sur 10 échantillons et présentés dans la Table B.4.

B.5.2 Prédiction des actions stratégiques des joueurs

Nous souhaitons capturer l'évolution des engagements sur les préférences au fur et à mesure du dialogue pour prédire le comportement stratégique des agents. Autre-

Type de res.	Précision	Rappel	F-mesure
Recevable	66,8	71,4	69,0
Donnable	62,6	59,7	61,1
Non-Don.	88,1	89,7	88,9
Non-Rec.	0	0	0
F-mesure moyenne			**54,8**
Accuracy			**67,4**

TABLE B.4 – Résultats pour la classification du type de ressource.

ment dit, nous souhaitons prédire parmi les 61 actions d'échange possibles, celle qui va être exécutée à la fin du dialogue de négociation. Les actions possibles sont des échanges qui varient sur le partenaire que le joueur dont c'est le jour de jeu choisi pour faire l'échange (3 options possibles pour un jeu à 4 joueurs), les ressources échangées (en supposant que les joueurs donne un type de ressource en échange d'un autre type, on obtient $5 \times 4 = 20$ possibilités) ou bien il n'y a pas d'échange. Au total, il y a donc $(3 \times 20) + 1 = 61$ actions possibles.

La prédiction des actions d'échange exécutées repose sur l'identification des préférences sur les ressources exprimées dans chaque UDE grâce à leur type (*Donnable*, etc.) associée à l'utilisation des actes de dialogue (*Offre*, *Acceptation*, etc.) pour modéliser l'évolution des préférences dans un modèle qui trace les préférences au fur et à mesure où elles sont exprimées. Toutes ces informations sont obtenues grâce aux sorties de nos classifieurs (voir Section B.5.1). Le modèle des préférences utilise les CP-nets (Boutilier et al., 2004) (voir Section B.1.2.2). Un CP-net est construit pour chaque agent et à la fin du dialogue de négociation les CP-nets obtenus sont utilisés pour prédire automatiquement l'action d'échange exécutée grâce à des principes de la théorie des jeux (Bonzon, 2007).

B.5.2.1 Prédire les échanges à partir des préférences

Dans notre corpus ***Settlers***, les préférences impliquent des quadruplets *(o, a, <r,q>)* où : *o* est le propriétaire de la préférence, *a* est son destinataire, *r* est la ressource et *q* sa quantité. Chaque variable des CP-nets correspond donc à un tel quadruplet associée aux valeurs possibles : *Donnable* (Don), *Non-Donnable* ($\overline{Don}$), *Recevable* (Rec) et *Non-Recevable* ($\overline{Rec}$).

Par exemple, l'énoncé *Est-ce que quelqu'un a de la laine pour de l'argile ?* exprime deux préférences : une pour recevoir de la laine, représentée par la variable

P_l = (A,Tous,<laine,1>) ; et étant donné cette préférence, une pour recevoir de l'argile, représentée par la variable P_a = (A,Tous,<argile,1>) (où A est le nom du joueur qui parle). Le CP-Net correspondant est présenté dans la Figure B.3.

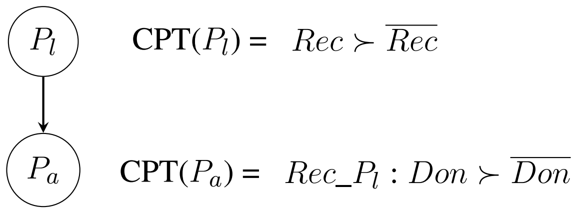


FIGURE B.3 – Un exemple de CP-net pour le corpus ***Settlers***.

Comme dit précédemment, la prédiction des actions d'échange exécutées utilise tout d'abord la sortie de notre premier classifieur pour identifier les préférences sur les ressources avec leur type ((*Donnable*, etc.) exprimées dans chaque UDE. Nous appliquons ensuite les règles présentées dans la Section B.4 pour construire dynamiquement le modèle des préférences avec les CP-nets. Nous n'utilisons pas directement la structure discursive puisqu'il n'existe pas encore de parseur discursif de la SDRT pour notre corpus ***Settlers***. Afin de proposer une méthode automatique de bout en bout, nous utilisons donc une équivalence entre les relations rhétorique (*Elaboration*, *Correction*, etc.) utilisées dans le Chapitre B.4 et les actes de dialogues que nous pouvons extraire automatiquement (*Offre*, *Acceptation*, etc.) (voir section B.5.1.1). Nous présentons très brièvement cette équivalence.

Les ***Offres*** ont un effet similaire à la relation d'*Elaboration* présentée en Section B.4.1.1. Les ***Contre-offres*** se comportent comme les *Offres* avec simplement un engagement partiel sur les préférences précédemment exprimées. Les ***Acceptations*** et les ***Refus*** se comportent comme les réponses à des QAP. Plus précisément, les *Acceptations* se comportent comme une réponse de type *Oui* quand les *Refus* correspondent à une réponse de type *Non*. Enfin, les UDE associées à la catégorie ***Autre*** ne nous intéressent pas car elles correspondent à des segments qui ne sont pas directement pertinents pour la négociation. Elles sont donc ignorées par la méthode.

A la fin du dialogue de négociation, nous prédisons quel échange exactement est réalisé (si un échange est effectivement fait) en vérifiant dans les CP-nets de chaque agent s'il y a des préférences complètes et réciproques. C'est à dire qu'on cherche si le CP-net de l'agent A exprime la préférence de donner à B la ressource r_1 en échange de r_2 tandis que celui de l'agent B exprime la préférence de donner à A la ressource r_2 en échange de r_1. S'il n'y a pas de correspondance entre les préférences de deux agents, alors on prédit qu'aucun n'échange n'est réalisé.

B.5.2.2 Exemple

Nous illustrons la méthode sur l'exemple traduit présenté dans la Table B.5 pour lequel nous détaillons la construction des CP-nets et la prédiction de l'action d'échange résultante.

ID	Acte	Texte	Joueur	Destinataire	Ressource
1	Offre	quelqu'un a de l'argile ?	Rainbow	Tous	Recevable (argile, ?)
2	Refus	Non, désolé	inca	Rainbow	
3	Refus	Pas pour le moment.	ariachiba	Rainbow	
4	Refus	j'ai besoin du mien	Kittles	Rainbow	Non-offrable (Anaphore, ?) Anaphore :(mine , argile)
5	Offre	quelqu'un a t-il du minerai ?	Rainbow	Tous	Recevable (minerai, ?)
6	Accept.	oui moi	Kittles	Rainbow	
7	Contre-of.	du minerai pour de la laine ?	Kittles	Rainbow	Donnable (minerai, ?) Recevable (laine, ?)
8	Accept.	ouais	Rainbow	Kittles	
9	Accept.	ok	Kittles	Rainbow	

TABLE B.5 – L'annotation stratégique d'un dialogue du corpus ***Settlers***.

π_1 L'UDE est une *Offre*, donc le CP-net de Rainbow est mis à jour par rapport au contenu de π_1.

CPT(R,Tous,<argile, ?>) = $Rec \succ \overline{Rec}$

π_2 C'est un *Refus*, donc nous mettons à jour le CP-net d'inca avec la négation des préférences exprimées dans l'offre de Rainbow.

CPT(I,R,<argile, ?>) = $\overline{Don} \succ Don$

π_3 Idem pour ariachiba.

CPT(A,R,<argile, ?>) = $\overline{Don} \succ Don$

π_4 Idem pour Kittles où les préférences exprimées dans l'UDE sont redondantes avec la négation des préférences de l'offre.

CPT(K,R,<argile, ?>) = $\overline{Don} \succ Don$

π_5 L'UDE est une *Offre*, donc le CP-net de Rainbow est tout d'abord mis à jour par rapport aux UDE précédentes (π_2 à π_4 depuis sa dernière prise de parole), puis par rapport au contenu de π_5. Donc nous obtenons :

CPT(R,Tous,<argile, ?>) = $Rec \succ \overline{Rec}$ ***(inactive)***
CPT(R,I,<argile, ?>) = $\overline{Rec} \succ Rec$

$CPT(R,A,<argile, ?>) = \overline{Rec} \succ Rec$
$CPT(R,K,<argile, ?>) = \overline{Rec} \succ Rec$
$CPT(R,Tous,<minerai, ?>) = \overline{Rec}(R,I,<argile, ?>) \wedge \overline{Rec}(R,A, <argile, ?>) \wedge \overline{Rec}(R,K,<argile, ?>) : Rec \succ \overline{Rec}$

L'introduction de la nouvelle préférence de recevoir du minerai est en conflit avec la préférence précédente de recevoir de l'argile. Donc la méthode ajoute une étiquette "inactive" à la CPT associée pour indiquer que la préférence est plus ancienne et qu'elle devra être ignorée si la préférence sur le minerai est satisfaite.

π_6 L'UDE est une *Acceptation*, donc le CP-net de Kittles est mis à jour en fonction des UDE précédentes (seulement π_5).[7]

$CPT(K,R,<minerai, ?>) = \overline{Don}(K,R,<argile, ?>) : Don \succ \overline{Don}$

π_7 L'UDE est une *Contre-offre*. Le CP-net est mis à jour seulement par rapport au contenu de l'UDE en cours puisque Kittles était la dernière personne à parler. Nous obtenons :

$CPT(K,R,<minerai, ?>) = \overline{Don}(K,R,<argile, ?>) : Don \succ \overline{Don}$
$CPT(K,R,<laine, ?>) = \overline{Don}(K,R,<argile, ?>) \wedge Don(K,R, <minerai, ?>) : Rec \succ \overline{Rec}$

π_8 L'UDE est une *Acceptation*, donc le CP-net de Rainbow est mis à jour par rapport aux UDE précédentes (π_6 et π_7) :

$CPT(R,K,<minerai, ?>) = \overline{Rec}(R,I,<argile, ?>) \wedge \overline{Rec}(R,A, <argile, ?>) \wedge \overline{Rec}(R,K,<argile, ?>) : Rec \succ \overline{Rec}$
$CPT(R,K,<laine, ?>) = \overline{Rec}(R,I,<argile, ?>) \wedge \overline{Rec}(R,A, <argile, ?>) \wedge \overline{Rec}(R,K,<argile, ?>) \wedge Rec(R,K,<minerai, ?>) : Don \succ \overline{Don}$

π_9 L'UDE est une *Acceptation* mais il n'y a rien de nouveau à mettre à jour.

A la fin du dialogue, les CP-nets des agents permettent de prédire (correctement) que Kittles donne du minerai à Rainbow en échange de laine.

B.5.2.3 Evaluation et résultats

Nous comparons notre méthode à quatre baselines.

La première baseline prédit un échange en fonction de la première *Offre* et de la dernière personne à faire une *Acceptation* (s'il n'y a pas d'*Acceptation* dans le

[7] Dans les CP-nets suivants, nous ne copions pas les CPT inactives et les CPT qui concernent des ressources *Non-Donnable* ou *Non-Recevable*.

dialogue alors la baseline prédit qu'il n'y a pas d'échanges. Ce sera similaire pour les trois autres baselines). Et si l'*Offre* ne précise pas une des ressources échangée alors elle est choisie aléatoirement (il en sera de même pour les autres baselines). Par exemple pour l'exemple de la Table B.5, la baseline prédit que Kittles donne de l'argile à Rainbow (ce qui est incorrect) en échange de quelque chose choisi aléatoirement (ce qui sera probablement incorrect).

La seconde baseline utilise la dernière *Offre* et la dernière personne à faire une *Acceptation*. Par exemple pour la Table B.5, la méthode prédit que Kittles donne du minerai à Rainbow (correct) pour quelque chose d'aléatoirement choisi (probablement incorrect).

La troisième baseline utilise la dernière *Offre* ou *Contre-offre*, selon laquelle est la dernière, et la dernière personne qui fait une *Acceptation*. Par exemple pour la Table B.5, la méthode prédit correctement que Kittles donne du minerai à Rainbow en échange de laine.

Et la quatrième baseline utilise une unification entre les *Offres* ou *Contre-offres* précédentes et courante pour mettre à jour les différents attributs de l'offre unifiée. La méthode prédit ensuite un échange en fonction de l'offre unifiée au moment de la dernière *Acceptation*. Pour la Table B.5, on obtient la même prédiction correcte que pour la troisième baseline. Mais cette dernière méthode est plus performante pour prédire les échanges corrects et complets comme dans les cas comme celui de l'exemple (B.4) (pour lequel la méthode prédit que Rainbow donne de la laine à Kittles pour du blé, plutôt que pour quelque chose d'aléatoire).

(B.4)	Rainbow :	j'ai besoin d'argile minerai ou blé
	Kittles :	j'ai du blé
	Rainbow :	j'peux te donner de la laine
	Kittles :	ok

Nous évaluons notre méthode sur les données présentées en Section B.2.1.2 : 254 dialogues au total puisque nous ignorons les dialogues ne contenant que des EDU de type *Autre*. 90 de ces dialogues se terminent par un échange et 2 se terminent par deux échanges. Une baseline aléatoire donne une accuracy de 1,6% (étant donné les 61 actions d'échange possible) et une baseline de fréquence (toujours prédire qu'il n'y a pas d'échange) donne une accuracy de 64,1%. La Table 6.4 présente les résultats pour la méthode et les quatre baselines.

Les résultats présentés dans cette table sont calculés à partir des données de l'annotation humaine plutôt que des sorties des classifieurs de la Section B.5.1 afin d'évaluer les modèles sans l'influence des erreurs des classifieurs du pré-traitement.

1^{ere} baseline : première *Offre*/dernière *Acceptation*					
VP	FP	FN	VN	PE	Accuracy
24	14	30	150	38	**68,0**
2^{eme} baseline : dernière *Offre*/dernière *Acceptation*					
VP	FP	FN	VN	PE	Accuracy
29	6	32	158	31	**73,0**
3^{eme} baseline : dernière *(Contre)Offre*/dernière *Acceptation*					
VP	FP	FN	VN	PE	Accuracy
39	4	23	160	30	**77,7**
4^{eme} baseline : *unification*					
VP	FP	FN	VN	PE	Accuracy
64	4	23	160	5	**87,5**
Notre méthode					
VP	FP	FN	VN	PE	Accuracy
75	4	15	160	2	**91,8**

TABLE B.6 – Résultats pour la prédiction des échanges.
VP, FP, FN, VN et PE sont les Vrais et Faux Positifs,
Faux et Vrais Négatifs et les Positifs Erronés.

Au contraire, dans la table B.7, nous présentons les résultats pour la méthode automatique de bout en bout (avec les prédictions d'échange faites à partie de la sortie des classifieurs). Le test de McNemar montre que notre méthode donne des résultats significativement meilleurs que ceux des baselines. Les Vrais Positifs (VP) concernent les cas où le modèle ne prédit pas seulement qu'un échange a eu lieu mais prédit aussi le bon partenaire et les bonnes ressources échangées (on ne tient pas compte de leur quantité) ; les Positifs Erronés (PE), au contraire, prédisent correctement qu'un échange a lieu mais ne prédisent pas le bon partenaire et/ou les bonnes ressources (donc les PE diminuent l'accuracy). Les Vrais Négatifs (VN) concernent les cas où le modèle prédit correctement qu'il n'y a pas d'échange (donc les VP et les VN contribuent à l'accuracy). Les Faux Positifs (FP) et les Faux Négatifs (FN) sont respectivement les prédictions incorrectes qu'il y a un échange, ou qu'il n'y a pas d'échange.

Cela n'apparait pas dans la Table 6.4, mais les trois premières baselines ont tendance à prédire des échanges avec des informations incomplètes avec souvent la nécessité de faire un choix aléatoire sur les ressources manquantes de l'*Offre* ou de la *Contre-offre*. Pour la première baseline, 34 exemples sont comme cela ; pour la deuxième et la troisième baselines, il y en a 32. Au contraire, ce problème ne concerne qu'un seul cas pour la quatrième baseline et aucun pour notre méthode. De plus, les trois premières baselines font souvent des prédictions erronées à propos du partenaire ou des ressources échangées parce qu'au contraire de notre méthode et de la quatrième baseline, elles ne suivent pas correctement l'évolution des échanges potentiels à travers la séquence d'offres et contre-offres.

Même si la quatrième baseline, qui utilise l'unification pour suivre le contenu de l'offre en cours, est assez intelligente et donne de bons résultats, elle obtient une accuracy significativement inférieure à celle de notre méthode. Un de ses problèmes majeurs est, qu'au contraire de notre méthode, elle ne perçoit pas bien l'engagement des joueurs sur l'offre en cours et repose uniquement sur la présence d'une *Acceptation* pour prédire s'il y a, ou non, un échange. Mais plusieurs exemples du corpus sont comme (B.5) dans lequel un échange est exécuté même s'il n'y a pas d'*Acceptation*, ce qui conduit à un Faux Négatif (FN) pour chacune des baselines.

(B.5)	Joel :	quelqu'un a du mouton ou de la laine
	Cardlinger :	aucun :(
	Joel :	je donnerai de l'argile ou du minerai
	Euan :	pas maintenant
	Jon :	de la laine pour de l'argile

(*Joel donne de l'argile à Jon et reçoit de la laine.*)

Notre évaluation montre donc que le raisonnement avec les CP-nets donne de meilleurs résultats que les quatre baselines qui ne capturent pas bien l'évolution des préférences.

4^{eme} baseline : *unification*					
VP	FP	FN	VN	PE	Accuracy
23	12	37	152	32	**68,4**
Notre méthode					
VP	FP	FN	VN	PE	Accuracy
34	10	43	154	15	**73,4**

TABLE B.7 – Résultats pour la prédiction automatique globale des échanges.

B.6 Conclusion

Nous avons proposé une méthode pour extraire et modéliser les préférences dans des conversations. Notre approche utilise des techniques du Traitement Automatique des Langues pour extraire automatiquement les préférences exprimées dans chaque segment de dialogue. Elle propose ensuite un ensemble de règles symboliques pour modéliser les préférences extraites dans des CP-nets qui permettent de suivre l'évolution au fur et à mesure que le dialogue progresse. Nous avons testé la méthode sur différents corpus de dialogues de négociation (coopératifs ou non).

Nous avons appliqué ce travail à une application concrète de prédiction des actions stratégiques dans le jeu *Les Colons de Catane*. L'extraction automatique des préférences et leur modélisation en CP-nets associées à des méthodes de raisonnement de la Théorie des Jeux permettent de prédire quels échanges seront réalisés ou non à la fin de chaque période de négociation. La méthode obtient de bons résultats, significativement meilleurs que ceux des quatre baselines de comparaison.

Dans le futur, nous souhaitons étendre la méthode avec un parseur automatique du discours. En effet, pour le moment, nous avons testé la méthode soit à partir de l'annotation discursive de la SDRT fournie par annotation manuelle (voir Section B.3), soit par une équivalence avec des actes de dialogues plus simple à prédire automatiquement (voir Section B.5). Mais pour la suite, nous prévoyons d'utiliser un parseur discursif pour obtenir automatiquement la structure de nos dialogues. Dans le cadre du projet européen STAC (ERC grant 269427), nous prévoyons d'adapter le parseur SDRT pour des documents en français de Muller et al. (2012) à notre corpus de dialogues ***Settlers*** et de réévaluer la méthode.

Nous voulons également tester la méthode sur un plus grand nombre de données dans des corpus plus variés. Dans une collaboration récente avec Diego Mollá et Abeed Sarker lors d'un séjour de trois mois à la Macquarie University de Sydney, nous avons commencé à étudier l'extraction des préférences dans un domaine complètement différent : les données médicales. Les premiers résultats obtenus montrent comment notre étude des préférences sur des dialogues de négociation est pertinente pour retrouver les recommandations dans un corpus de questions médicales initialement construit pour faire du résumé de textes (Mollá, 2010; Mollá and Santiago-Martínez, 2011; Sarker et al., 2013).

A plus long terme, nous prévoyons d'appliquer la méthode à des systèmes d'élicitaion des préférences afin de proposer un outil dans lequel les utilisateurs puissent exprimer leurs préférences en langue naturelle. Mais plutôt que de traduire les préférences en requêtes SQL comme dans le système *ExpertClerk* (Shimazu, 2001) (voir Section B.1), nous pouvons imaginer un système de dialogue (López-Cózar Delgado and Araki, 2005) qui raisonne directement sur les préférences exprimées par l'utilisateur et lui permette de les raffiner en langue naturelle.

Bibliography

[Afantenos et al., 2012a] Stergos Afantenos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Péry-Woodley, Laurent Prévot, Josette Rebeyrolle, Ludovic Tanguy, Marianne Vergez-Couret, and Laure Vieu. 2012a. An empirical resource for discovering cognitive principles of discourse organisation: the annodis corpus. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC'12)*, pages 2727–2734. European Language Resources Association (ELRA).

[Afantenos et al., 2012b] Stergos Afantenos, Nicholas Asher, Farah Benamara, Anaïs Cadilhac, Cédric Dégremont, Pascal Denis, Markus Guhe, Simon Keizer, Alex Lascarides, Oliver Lemon, Philippe Muller, Soumya Paul, Verena Rieser, and Laure Vieu. 2012b. Developing a corpus of strategic conversation in the settlers of catan. In *Proceedings of the 1st Workshop on Games and NLP (GAMNLP-12)*.

[Alchourrón et al., 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. 1985. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530.

[Aloysius et al., 2006] John A. Aloysius, Fred D. Davis, Darryl D. Wilson, A. Ross Taylor, and Jeffrey E. Kottemann. 2006. User acceptance of multi-criteria decision support systems: The impact of preference elicitation techniques. *European Journal of Operational Research*, 169(1):273–285.

[Amato and Shani, 2010] Christopher Amato and Guy Shani. 2010. High-level reinforcement learning in strategy games. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10)*, pages 75–82.

[Ariely, 2008] Dan Ariely. 2008. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. HarperCollins.

[Arora and Allenby, 1999] Neeraj Arora and Greg M. Allenby. 1999. Measuring the influence of individual preference structures in group decision making. *Journal of Marketing Research*, 36:476–487.

[Artstein and Poesio, 2008] Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

[Asher and Lascarides, 2003] Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

[Asher and Quinley, 2011] Nicholas Asher and Jason Quinley. 2011. Begging questions, their answers and basic cooperativity. In *Proceedings of the 8th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS) - Workshop at JSAI-isAI*.

[Asher et al., 2010] Nicholas Asher, Elise Bonzon, and Alex Lascarides. 2010. Extracting and modelling preferences from dialogue. In *IPMU*, pages 542–553.

[Asher et al., 2013] Nicholas Asher, Cédric Dégremont, and Soumya Paul. 2013. A logic of sights. forthcoming in *Logic and Computation*.

[Asher, 1987] Nicholas Asher. 1987. A typology for attitude verbs and their anaphoric properties. *Linguistics and Philosophy*, 10:125–197.

[Bacchus and Grove, 1995] Fahiem Bacchus and Adam Grove. 1995. Graphical models for preference and utility. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI 1995)*. Morgan Kaufmann Publishers Inc.

[Baldridge and Lascarides, 2005a] Jason Baldridge and Alex Lascarides. 2005a. Annotating discourse structures for robust semantic interpretation. In *Proceedings of the 6th IWCS*.

[Baldridge and Lascarides, 2005b] Jason Baldridge and Alex Lascarides. 2005b. Probabilistic head-driven parsing for discourse structure. In *Proceedings of CoNLL*.

[Bellman, 1957] Richard Bellman. 1957. A markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684.

[Benamara et al., 2011] Farah Benamara, Baptiste Chardon, Yannick Mathieu, and Vladimir Popescu. 2011. Towards context-based subjectivity analysis. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1180–1188, Chiang Mai, Thailand.

[Benferhat et al., 1993] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. 1993. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the 13th international joint conference on Artifical Intelligence (IJCAI'93)*, pages 640–645.

[Bigot et al., 2013] Damien Bigot, Hélène Fargier, Jérôme Mengin, and Bruno Zanuttini. 2013. Probabilistic conditional preference networks. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, pages 72–81. Association for Uncertainty in Artificial Intelligence.

[Bistarelli et al., 1999] Stefano Bistarelli, Ugo Montanari, Francesca Rossi, Thomas Schiex, Gérard Verfaillie, and Hélène Fargier. 1999. Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison. *Constraints*, 4(3):199–240.

[Bonzon, 2007] Elise Bonzon. 2007. *Modélisation des interactions entre agents rationnels : les jeux booléens*. PhD thesis, Université Paul Sabatier, Toulouse.

[Booth et al., 2010] Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera. 2010. Learning conditionally lexicographic preference relations. In *Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 269–274. IOS Press.

[Boutilier et al., 1999] Craig Boutilier, Ronen I. Brafman, Holger H. Hoos, and David Poole. 1999. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 71–80, Stockholm, Sweden. Morgan Kaufmann Publishers Inc.

[Boutilier et al., 2001] Craig Boutilier, Fahiem Bacchus, and Ronen I. Brafman. 2001. UCP-networks: A directed graphical representation of conditional utilities. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence (UAI-01)*, pages 56–64. Morgan Kaufmann Publishers Inc.

[Boutilier et al., 2004] Craig Boutilier, Craig Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. 2004. CP-nets: A tool for representing and reasoning with conditional *ceteris paribus* preference statements. *Journal of Artificial Intelligence Research (JAIR)*, 21:135–191.

[Bouveret et al., 2009] Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. 2009. Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 67–72. Morgan Kaufmann Publishers Inc.

[Bradley, 2007] Richard Bradley. 2007. The kinematics of belief and desire. *Synthese*, 156(3):513–535.

[Brafman and Domshlak, 2002] Ronen I. Brafman and Carmel Domshlak. 2002. Introducing variable importance tradeoffs into CP-nets. In *Proceedings of the eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 69–76. Morgan Kaufmann Publishers Inc.

[Brafman and Domshlak, 2009] Ronen I. Brafman and Carmel Domshlak. 2009. Preference handling – an introductory tutorial. *AI Magazine*, 30(1):58–86.

[Brainov, 2000] Sviatoslav Brainov. 2000. The role and the impact of preferences on multiagent interaction. In *6th International Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL)*, pages 349–363. Springer-Verlag.

[Browne et al., 2012] Cameron Browne, Edward Powley, Daniel Whitehouse, Simon Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–49.

[Burges, 1998] Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

[Burke, 2000] Robin Burke. 2000. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Science*, volume 69. Marcel Dekker. Inc.

[Cadilhac et al., 2011] Anaïs Cadilhac, Nicholas Asher, Farah Benamara, and Alex Lascarides. 2011. Commitments to preferences in dialogue. In *Special Interest Group on Discourse and Dialogue (SIGdial)*, pages 204–215. Association for Computational Linguistics (ACL).

[Cadilhac et al., 2012a] Anaïs Cadilhac, Nicholas Asher, and Farah Benamara. 2012a. Annotating preferences in chats for strategic games. In *the 6th Linguistic Annotation Workshop (The LAW VI)*, pages 139–143. Association for Computational Linguistics (ACL).

[Cadilhac et al., 2012b] Anaïs Cadilhac, Nicholas Asher, and Farah Benamara. 2012b. Annotating preferences in negotiation dialogues. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pages 105–113. Association for Computational Linguistics (ACL).

[Cadilhac et al., 2012c] Anaïs Cadilhac, Nicholas Asher, Farah Benamara, Vladimir Popes cu, and Mohamadou Seck. 2012c. Preference extraction from negotiation dialogues. In *European Conference on Artificial Intelligence (ECAI)*, pages 211–216. IOS Press.

[Cadilhac et al., 2012d] Anaïs Cadilhac, Farah Benamara, Vladimir Popes cu, Nicholas Asher, and Mohamadou Seck. 2012d. Extraction de préférences à partir de dialogues de négociation. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 343–350. ATALA/AFCP.

[Cadilhac et al., 2013] Anaïs Cadilhac, Nicholas Asher, Farah Benamara, and Alex Lascarides. 2013. Grounding strategic conversation: Using negotiation dialogues to predict trades in a win-lose game. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 357–368. Association for Computational Linguistics.

[Cadilhac et al., submitted] Anaïs Cadilhac, Nicholas Asher, Farah Benamara, and Alex Lascarides. submitted. Preference change.

[Charniak, 2000] Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*, pages 132–139.

[Chen and Pu, 2004] Li Chen and Pearl Pu. 2004. Survey of preference elicitation methods. Technical report, Swiss Federal Institute of Technology in Lausanne (EPFL).

[Chen and Pu, 2007] Li Chen and Pearl Pu. 2007. Hybrid critiquing-based recommender systems. pages 22–31.

[Core and Allen, 1997] Mark G. Core and James F. Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*.

[Coste-Marquis et al., 2004] Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore, and Pierre Marquis. 2004. Expressive power and succinctness of propositional languages for preference representation. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Proceedings of the 9th International Conference on Principles of Knowledge Representation ans Reasonning (KR'04)*, pages 203–212. AAAI Press.

[de Gemmis et al., 2009] Marco de Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2009. Preference learning in recommender systems. In *Preference Learning (PL-09) ECML/PKDD-09 Workshop*.

[Dereszynski et al., 2011] Ethan W. Dereszynski, Jesse Hostetler, Alan Fern, Thomas G. Dietterich, Thao-Trang Hoang, and Mark Udarbe. 2011. Learning probabilistic behavior models in real-time strategy games. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 20–25. The AAAI Press.

[Dietterich, 1998] Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

[Domshlak et al., 2011] Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. 2011. Preferences in AI: An overview. *Artificial Intelligence, Representing, Processing, and Learning Preferences : Theoretical and Practical Challenges*, 175(7-8):1037–1052.

[Dubois et al., 1994] Didier Dubois, Jérôme Lang, and Henri Prade. 1994. Handbook of logic in artificial intelligence and logic programming (vol. 3). chapter Possibilistic logic, pages 439–513. Oxford University Press, Inc.

[Dubois et al., 2006] Didier Dubois, Souhila Kaci, and Henri Prade. 2006. Approximation of conditional preferences networks "CP-nets" in possibilistic logic. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'06)*.

[Dupin de Saint Cyr-Bannay et al., 1994] Florence Dupin de Saint Cyr-Bannay, Jérôme Lang, and Thomas Schiex. 1994. Penalty logic and its link with dempster-shafer theory. In Ramon Lopez de Mantaras and David Poole, editors, *Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 204–211. Morgan Kaufmann Publishers Inc.

[Ehlen and Johnston, 2013] Patrick Ehlen and Michael Johnston. 2013. A multimodal dialogue interface for mobile local search. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, pages 63–64. ACM.

[Erol et al., 1994] Kutluhan Erol, James Hendler, and Dana S. Nau. 1994. HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94*, volume 2, pages 1123–1128. AAAI Press.

[Fernández et al., 2005] Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2005. Using machine learning for non-sentential utterance classification. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 77–86.

[Flach and Matsubara, 2007] Peter Flach and Edson Takashi Matsubara. 2007. A simple lexicographic ranker and probability estimator. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 575–582. Springer-Verlag.

[Fraser, 1994] Niall M. Fraser. 1994. Ordinal preference representations. *Theory and Decision*, 36(1):45–67.

[Fürnkranz and Hüllermeier, 2011] Johannes Fürnkranz and Eyke Hüllermeier, editors. 2011. *Preference Learning*. Springer.

[Ganapathibhotla and Liu, 2008] Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 241–248, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Gärdenfors and Makinson, 1988] Peter Gärdenfors and David Makinson. 1988. Revisions of knowledge systems using epistemic entrenchment. In Moshe Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 83–95. Morgan Kaufmann.

[Gilpin and Sandholm, 2007] Andrew Gilpin and Tuomas Sandholm. 2007. Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5):1–32.

[Gonzales and Perny, 2004] Christophe Gonzales and Patrice Perny. 2004. GAI networks for utility elicitation. In *Proccedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR'04*, pages 224–234. AAAI Press.

[Grüne-Yanoff and Hansson, 2009a] Till Grüne-Yanoff and Sven Ove Hansson. 2009a. Preference change: An introduction. In Till Grüne-Yanoff and Sven Ove Hansson, editors, *Preference Change: Approaches from Philosophy, Economics and Psychology*, pages 1–26. Springer Netherlands.

[Grüne-Yanoff and Hansson, 2009b] Till Grüne-Yanoff and Sven Ove Hansson. 2009b. *Preference Change: Approaches from Philosophy, Economics and Psychology*. Springer Netherlands.

[Hansson and Grüne-Yanoff, 2006] Sven Ove Hansson and Till Grüne-Yanoff. 2006. Preferences. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition.

[Hansson, 1995] Sven Ove Hansson. 1995. Changes in preference. *Theory and Decision*, 38(1):1–28.

[Hansson, 2006] Sven Ove Hansson. 2006. Logic of belief revision. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2011 edition.

[Harsanyi, 1977] John C. Harsanyi. 1977. *Rational behavior and bargaining aquilibrium in games and social situations*. Cambridge University Press.

[Hausman, 2000] Daniel M. Hausman. 2000. Revealed preference, belief, and game theory. *Economics and Philosophy*, 16(01):99–115.

[Hill, 2009] Brian Hill. 2009. Three analyses of sour grapes. In Till Grüne-Yanoff and Sven Ove Hansson, editors, *Preference Change: Approaches from Philosophy, Economics and Psychology*, pages 27–56. Springer Netherlands.

[Ivanovic, 2008] Edward Ivanovic. 2008. Automatic instant messaging dialogue using statistical models and dialogue acts. Master's thesis, Faculty of Engineering, Computer Science and Software Engineering, University of Melbourne.

[Jia et al., 2009] Lifeng Jia, Clement T. Yu, and Weiyi Meng. 2009. The effect of negation on sentiment analysis and retrieval effectiveness. In *CIKM*, pages 1827–1830.

[Jindal and Liu, 2006a] Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR'06)*, pages 244–251. ACM.

[Jindal and Liu, 2006b] Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of the 21st national conference on Artificial intelligence (AAAI'06)*, pages 1331–1336. AAAI Press.

[Joty et al., 2011] Shafiq Joty, Giuseppe Carenini, and Chin-Yew Lin. 2011. Unsupervised modeling of dialog acts in asynchronous conversations. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1807–1813. AAAI Press.

[Kaci and Prade, 2008] Souhila Kaci and Henri Prade. 2008. Mastering the processing of preferences by using symbolic priorities in possibilistic logic. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 376–380. IOS Press.

[Kaci, 2011] Souhila Kaci. 2011. *Working with Preferences: Less Is More*. Cognitive Technologies. Springer.

[Kahneman and Tversky, 1979] Daniel Kahneman and Amos Tversky. 1979. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292.

[Keizer et al., 2002] Simon Keizer, Rieks op den Akker, and Anton Nijholt. 2002. Dialogue act recognition with bayesian networks for dutch dialogues. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, pages 88–94. Association for Computational Linguistics.

[Kim et al., 2010] Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871.

[Kim et al., 2012] Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2012. Classifying dialogue acts in multi-party live chats. In *26th Pacific Asia Conference on Language,Information and Computation*, pages 463–472.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430. Association for Computational Linguistics.

[Krulwich, 1997] Bruce Krulwich. 1997. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45.

[Lafage and Lang, 2000] Céline Lafage and Jérôme Lang. 2000. Logical representation of preferences for group decision making. In *Proceeding of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 457–468.

[Lang and van der Torre, 2008] Jérôme Lang and Leon van der Torre. 2008. From belief change to preference change. In *Proceedings of the Eighteenth European Conference on Artificial Intelligence (ECAI 2008)*, pages 351–355.

[Li et al., 2010] Junhui Li, Guodong Zhou, Hongling Wang, and Qiaoming Zhu. 2010. Learning the scope of negation via shallow semantic parsing. In *COLING*, pages 671–679.

[Liu, 2008] Fenrong Liu. 2008. *Changing for the Better: Preference Dynamics and Agent Diversity*. Ph.D. thesis, Institute for Logic, Language and Computation (ILLC), University of Amsterdam.

[Liu, 2010] Bing Liu. 2010. Sentiment analysis and subjectivity. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group.

[López-Cózar Delgado and Araki, 2005] Ramón López-Cózar Delgado and Masahiro Araki. 2005. *Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment*. Wiley.

[Miller et al., 2003] Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. 2003. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces (IUI'03)*, pages 263–266. ACM.

[Mollá and Santiago-Martínez, 2011] Diego Mollá and María Elena Santiago-Martínez. 2011. Development of a corpus for evidence based medicine summarisation. In *Proceedings of Australasian Language Technology Association Workshop*, pages 86–94.

[Mollá, 2010] Diego Mollá. 2010. A corpus for evidence based medicine summarisation. In *Proceedings of Australasian Language Technology Workshop*, pages 76–80.

[Muller et al., 2012] Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1883–1900. Association for Computational Linguistics (ACL).

[Osborne and Rubinstein, 1994] Martin J. Osborne and Ariel Rubinstein. 1994. *A Course in Game Theory*. The MIT Press.

[Pazzani and Billsus, 1997] Michael Pazzani and Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331.

[Pazzani, 1999] Michael J. Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408.

[Pearl, 1990] Judea Pearl. 1990. System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the 3rd conference on Theoretical aspects of reasoning about knowledge (TARK'90)*, pages 121–135.

[Pommeranz et al., 2012] Alina Pommeranz, Joost Broekens, Pascal Wiggers, Willem-Paul Brinkman, and Catholijn M. Jonker. 2012. Designing interfaces for explicit preference elicitation: a user-centered investigation of preference representation and elicitation process. *User Modeling and User-Adapted Interaction*, 22:357–397.

[Pu and Chen, 2008] Pearl Pu and Li Chen. 2008. User-involved preference elicitation for product search and recommender systems. *AI Magazine*, 29(4):93–103.

[Resnick et al., 1994] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (CSCW'94)*, pages 175–186. ACM Press.

[Rossi et al., 2008] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2008. Preferences in constraint satisfaction and optimization. *AI Magazine*, 29(4):58–68.

[Sarker et al., 2013] Abeed Sarker, Diego Mollá, and Cécile Paris. 2013. An approach for query-focused text summarisation for evidence based medicine. In *Proceedings of the 14th Conference on Artificial Intelligence in Medicine (AIME 2013)*, pages 295–304.

[Savage, 1954] Leonard J. Savage. 1954. *The Foundations of Statistics*. John Wiley.

[Schadd et al., 2007] Frederik Schadd, Sander Bakkes, and Pieter Spronck. 2007. Opponent modeling in real-time strategy games. In *Games and Simulation GAMEON*, pages 61–68.

[Schulz, 2007] Katrin Schulz. 2007. *Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals*. PhD thesis, ILLC.

[Shimazu et al., 1992] Hideo Shimazu, Seigo Arita, and Yosuke Takashima. 1992. Design tool combining keyword analyzer and case-based parser for developing natural language database interfaces. In *Proceedings of the 15th International Conference on Computational linguistics (COLING-92)*, volume 2, pages 735–741. Association for Computational Linguistics.

[Shimazu, 2001] Hideo Shimazu. 2001. ExpertClerk : Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, volume 2, pages 1443–1448. Morgan Kaufmann Publishers Inc.

[Shoham and Leyton-Brown, 2009] Yoav Shoham and Kevin Leyton-Brown. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

[Sidner, 1994] Candace L. Sidner. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of the 12th national conference on Artificial intelligence*, pages 814–819. American Association for Artificial Intelligence.

[Simon, 1955] Herbert A. Simon. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118.

[Spohn, 2009] Wolfgang Spohn. 2009. Why the received models of considering preference change must fail. In Till Grüne-Yanoff and Sven Ove Hansson, editors, *Preference Change: Approaches from Philosophy, Economics and Psychology*, pages 109–121. Springer Netherlands.

[Stolcke et al., 2000] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. In *Computational Linguistics*, volume 26, pages 339–373. The MIT Press.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. The MIT Press.

[Szita et al., 2010] István Szita, Guillaume Chaslot, and Pieter Spronck. 2010. Monte-carlo tree search in settlers of catan. In *Advances in Computer Games*, pages 21–32. Springer Berlin Heidelberg.

[Wahlster, 2000] Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.

[Wellman and Doyle, 1991] Michael P. Wellman and Jon Doyle. 1991. Preferential semantics for goals. In *Proceedings of the Ninth National Conference on Artificial intelligence (AAAI-91)*, pages 698–703. AAAI Press.

[Wellner et al., 2006] Ben Wellner, James Pustejovsky, Catherine Havasi, Anna Rumshisky, and Roser Saurí. 2006. Classification of discourse coherence relations: an exploratory study using multiple knowledge sources. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue (SIGdial'06)*, pages 117–125. Association for Computational Linguistics.

[Wu et al., 2002] Tianhao Wu, Faisal M. Khan, Todd A. Fisher, Lori A. Shuler, and William M. Pottenger. 2002. Posting act tagging using transformation-based learning. In *Foundations of Data Mining and knowledge Discovery*, pages 319–331. Springer.

[Yong and Xinlin, 2012] Zhao Yong and Wu Xinlin. 2012. Intrinsic preferences, revealed preferences and bounded rational decisions. *Systems Research and Behavioural Science*.