Preference Handling in Decision-Making Problems

Namrata Patel

Collège Doctoral du Languedoc-Roussillon

THÈSE

Pour obtenir le grade de Docteur

Délivré par l'UNIVERSITÉ DE MONTPELLIER

Préparée au sein de l'école doctorale I2S
Et de l'unité de recherche LIRMM

Spécialité : INFORMATIQUE

Présentée par Namrata PATEL

MISE EN OEUVRE DES PREFERENCES DANS LES PROBLEMES DE DECISION

Soutenue le 7 octobre 2016 devant le jury composé de :

Mme. Souhila KACI	Prof.	Univ. Montpellier	Directeur
M. Roland DUCOURNAU	Prof.	Univ. Montpellier	Co-Directeur
M. Nic WILSON	SRF	Univ. College Cork	Rapporteur
M. Farid NOUIOUA	MdC	Univ. Aix-Marseille	Examinateur
M. Nadjib LAZAAR	MdC	Univ. Montpellier	Examinateur

Et présidé par :

M. Jérôme LANG	DR CNRS	Univ. Paris Dauphine	Rapporteur

I offer this work to the Mother and Sri Aurobindo.

Acknowledgements

The present work has been the fruit of a long and passionate endeavour, culminating in the sweet release of a happy sense of accomplishment. Doctoral research, thanks to its length and the personal involvement that it requires, has the unique quality of providing a wealth of experiences and lessons to learn, both on the academic and personal fronts. In my case, these have greatly enriched my personality and I am grateful to have had this opportunity. None of this would have been possible without the unrelenting support of my family, friends, colleagues and most importantly Aurijoy, who had to brave it all for the past three years.

My first acknowledgement goes to my thesis supervisors Souhila Kaci and Roland Ducournau, without whom this work could never have reached its present form. I am particularly grateful to Marianne Huchard, who in her capacity as head of the entire doctoral unit, has had the time and energy to actively invest herself in the welfare of every PhD student.

In helping me ripen my thought, the academic support and encouragement from each member of my research team headed by Marie-Laure Mugnier have been exceptional and invaluable. Equally helpful have been the discussions with, and constant guidance from, my former teachers Violaine Prince, Mathieu Roche, Madalina Croitoru and last but certainly not the least, Jean-Philippe Prost.

I must also add here that I deeply appreciate the lab. support staff, every single one of them, for their dedicated involvement in my personal well-being. Foremost among these is Cécile Lukasik, who now knows me better than I do myself. Nicolas Serrurier, the stalwart supporter of the PhD students' cause, has been incredible during my entire stay, going so often out of his way to help a student in need. And thank you so much Laurie Lavernhe, for your unforgettable friendship.

A final acknowledgement goes to Jérôme Lang, Nic Wilson, Farid Nouioua and Nadjib Lazaar for honouring me by being a part of the jury for my public defence.

Résumé

Il y a une forte croissance, à nos jours, de «services» intelligents proposés aux clients sur les plates-formes de commerce électronique, destinés à une assistance personnalisée. L'étude de préférences a suscité un grand intérêt dans ce contexte, grâce à leur utilisation dans la résolution de problèmes liés à la prise de décision. En effet, la recherche sur les préférences en intelligence artificielle (IA) propose différentes manières d'aborder ce problème : de l'acquisition des préférences à leur représentation formelle et, éventuellement, à leur gestion suivant plusieurs méthodes de raisonnement. Dans cette thèse, nous adressons la problématique de la mise en œuvre de préférences comparatives pour l'aide à la décision par le développement d'un système interactif «intelligent» de recommandations personnalisées. Nous suivons une tendance récente, et le concevons sur une base de considérations psychologiques, linguistiques et personnelles. Nous contribuons ainsi aux domaines suivants de préférences en IA : (1) leur acquisition, (2) leur représentation, et (3) leur mise en œuvre. Nous examinons d'abord un goulot d'étranglement dans l'acquisition de préférences et proposons une méthode d'acquisition de préférences exprimées en langage naturel (LN), qui permet leur représentation formelle en tant que préférences comparatives. Nous étudions ensuite les aspects théoriques de la représentation et du raisonnement avec les préférences comparatives pour aide à la décision. Finalement, nous décrivons notre outil de recommandations qui utilise : (1) une base de données de produits qualifiée par une analyse de critiques d'utilisateurs, (2) une approche interactive pour guider les utilisateurs à exprimer leurs préférences comparatives, et (3) un moteur de raisonnement qui manipule ces préférences afin de proposer une recommandation basée sur les préférences de l'utilisateur.

Abstract

Intelligent 'services' are increasingly used on e-commerce platforms to provide assistance to customers. In this context, preferences have gained rapid interest for their utility in solving problems related with decision making. Research on preferences in artificial intelligence (AI) has shed light on various ways of tackling this problem, ranging from the acquisition of preferences to their formal representation and eventually their proper manipulation. Following a recent trend of stepping back and looking at decision-support systems from the user's point of view, i.e. designing them on the basis of psychological, linguistic and personal considerations, we take up the task of developing an"intelligent" tool which uses comparative preference statements for personalised decision support. We tackle and contribute to different branches of research on preferences in AI: (1) their acquisition (2) their formal representation and (3) their implementation. We first address a bottleneck in preference acquisition by proposing a method of acquiring user preferences, expressed in natural language (NL), which favours their formal representation and further manipulation. We then focus on the theoretical aspects of handling comparative preference statements for decision support. We finally describe our tool for product recommendation that uses: (1) a review-based analysis to generate a product database, (2) an interactive preference elicitation unit to guide users to express their preferences, and (3) a reasoning engine that manipulates comparative preference statements to generate a preference-based ordering on outcomes as recommendations.

Contents

Introduction

Decision making is an intrinsic part of human existence, and central to its accomplishment is the notion of preference. It is therefore no surprise that the study of preferences is intrinsic to research ranging from philosophy and psychology to economics and since the advent of computer science, to artificial intelligence (AI). The latter is exponentially gaining in importance, as technology makes the virtual world more and more real to us, and our cognitive capacities fail to keep up. We now have various decision-support systems such as web-based recommender systems, personal assistants, targeted advertising, etc. to simplify our daily life, and others designed to provide automated planning, scheduling and decision making in autonomous systems, such as NASA's Mars Exploration Rovers.

All this progress is the fruit of a very interesting point in time for AI research: it is poised in a dynamic equilibrium between theoretical and practical advances, caused by our readiness to seek the internet to assist us in all our activities. The practical advances keep our interest alive and the data we generate becomes the testing ground for further advances. It is a time brimming with significant breakthroughs, where years of theoretical research finally bear fruit in our everyday life.

Using this as a springboard, this thesis explores preference handling in theoretical AI research and its practical applications to personalised decision support.

Problem Statement

Intelligent 'services' are increasingly used on e-commerce platforms to provide assistance to customers. In this context, preferences have gained rapid interest for their utility in solving problems related to decision making. Research on preferences in AI has shed light on various ways of tackling this problem, right from the acquisition of preferences to their formal representation and eventually their proper manipulation. Numerous preference elicitation methods developed in the literature are now employed in intelligent services such as recommender systems. These have been selected for their ability to adapt to the existing paradigm in recommendation algorithms.

There is, however, a recent trend of stepping back and looking at such decision support systems from the user's point of view, i.e. designing them on the basis of psychological, linguistic and personal considerations. Seen in this light, there are several existing preference formalisms which are well-suited to personalised decision support, and remain still to be exploited in real-world applications. We follow this trend and carve the way for one such formalism, that of *comparative preference statements*, to make its journey from the abstract to the concrete.

By taking up the task of developing an "intelligent" tool which uses this less explored theme for personalised decision support, we tackle and contribute to different branches of research on preferences in AI: (1) their acquisition, (2) their formal representation and manipulation and (3) their implementation. Our approach towards the different studies goes beyond the scope of an isolated project culminating in the development of a single tool: we conduct them as work that contributes to the research and development community, opening up the opportunity for other applications to be built upon it.

Research Methodology

We begin our study with an in-depth analysis of research on preferences, making an educated choice about the formal representation language and its accompanying reasoning algorithms which would respond best to assisting users in decision-making. We then look at some of the state-of-the-art preference-based decision support systems to determine how best to implement a tool based on this language. This chalks out the research objectives for the present thesis.

Our first objective concerns the acquisition of preferences which includes (1) addressing an existing bottleneck by proposing a method of eliciting user preferences, expressed in natural language (NL), which favours their formal representation and further manipulation; (2) testing the feasibility of this method using a proof of concept experiment, thereby (3) constructing a corpus of preference expressions and an accompanying lexicon of preference terminology.

The next portion of our study focusses on the theoretical aspects of handling comparative preference statements for decision support. Considering that in practice this requires acquiring preferences expressed by a user, it would be useful to know how best to exploit the expressivity of the theoretical construct. This requires a thorough understanding of the very nature and behaviour of comparative preference statements. We therefore take up existing work on the topic and analyse it w.r.t. some of the basic principles that govern preference logics in general to support our intuition behind using this formalism.

In the final part of the study, we work out how all of the above can come together in an intelligent tool, capable of performing personalised decision support. We first design an interactive module for preference elicitation which uses statistically-driven methods in information retrieval to minimise user interaction, without losing out on expressivity. We then focus on the design of the core of our system: the reasoning engine. Our reasoning engine computes recommendations for the user, and is entirely based on the theoretical research on comparative preference statements. We complete our study by implementing the proposed framework in a specific scenario, discussing its performance and adherence to the theory's predictions.

Thesis Structure

Addressing the fundamentally multidisciplinary nature of preferences, we begin our thesis with a literature review in chapter 1, presenting a broad outline of their diverse involvement in decision support. Our own contributions being both theoretical and practical by nature, we adopt an 'inch-deep-mile-wide' outlook to reveal the intermingling between these, without going into the nitty-gritty of either aspect. This reveals the different research tracks of preference handling in decision support, laying out the motivations for our contributions. Chapters 2, 3 and 4 then use the 'mile-deep-inch-wide' approach on each individual contribution, allowing the reader to plunge into each without losing track of the unifying factor.

Chapter 1 In chapter 1 we follow the history and background of research on preferences and their involvement in decision-support. This includes the mathematical and logical foundations of representing and reasoning with preferences, along with an overview of some well-known preference formalisms in AI research. We then move on to exploring existing techniques in acquiring preferences, revealing a commonly-known bottleneck in this field. We conclude with a survey of the use of preferences in decision-support systems, focussing on how the formalisms visited have been incorporated into them.

Acquainting the reader with the existing work done on the topic, we then point out the areas we seek to contribute to, highlighting our choices with current research trends and motivations.

Chapter 2 Our first contribution addresses the bottleneck in preference acquisition that was revealed in chapter 1. We propose a method of acquiring user preferences, expressed in natural language (NL), which favours their formal representation and further manipulation using algorithms developed in previous research. In particular, we investigate expressions which could be adapted to *comparative preference statements*, since they offer an intuitive and natural way to represent user preferences and lead to many types of preference representation languages. Moreover, they can be further defined using different preference semantics (we call these 'AI preference semantics' to avoid confusion with the term 'semantics' in the linguistic sense), which lend a greater depth to these preferences. This is because each AI preference semantics offers a different way of ordering the outcomes that satisfy the given preferences. Our aim therefore is to acquire NL preferences that reflect the AI preference semantics that could be associated with them.

Our approach is to develop a protocol for preference elicitation and to build the linguistic resources it requires. These resources must not only capture NL preference expressions but also match them with the different AI preference semantics, which lead to distinct solving procedures. We design our study based on the following key questions: (1) Are preference linguistic patterns different from opinion expressions, when faced with AI preference semantics theories? (2) Does retrieving them require specific corpora, i.e. dialogue corpora since elicitation is a dynamic process, and if so, what are the linguistic clues denoting preference expressions? (3) Can natural language processing help in improving the elicitation process by increasing the accuracy of the interaction with the user?

We then test our protocol by means of a crowd-sourcing experiment which serves as a proof of concept, thereby providing a concrete link between natural language expressions and research in preferences in artificial intelligence. The linguistic resources it requires are built using two constructs: (1) a preference lexicon with a distinctive sorting method using Formal Concept Analysis (FCA) that maintains its semantic classification, and (2) *preference templates* which describe structural patterns using words from the lexicon that identify NL preference expressions and distinguish AI preference semantics. Through the results of our crowd-sourcing experiment, we have built a corpus which contains authentic user preferences in natural language corresponding to comparative statements and their associated semantics in artificial intelligence.

Chapter 3 The results of our first contribution show that comparative preference statements are a well-suited formalism for personalised decision support, by way of their (1) proximity to the intuitive way in which we express preferences and their (2) expressivity in reflecting the nuances of reasoning about preferences. Our second contribution addresses the theoretical construct of comparative preference statements and their associated semantics. We look at their origins, associated reasoning mechanisms, and make a deeper analysis about their behaviour.

We begin with a rigorous treatment of the formulation of comparative preference statements. This means going back to the mathematical modelling of preferences and building up our theory from there to the formulation of comparative preference statements. We discuss the different semantics defined for comparative preference statements and the pitfalls and advantages of using each one of them.

Next, we discuss the task of computing preference relations induced by sets of comparative preference statements and one or several semantics. We explain the existing reasoning mechanisms associated with these statements, and present algorithms developed in previous research for this purpose.

We conclude the chapter with a postulate-based analysis of these statements. Our selection of postulates is motivated by properties that could optimise the decision-making process (i.e. inferring new preferences from previously known preferences). Our analysis then consists of examining the affects of preference semantics on comparative preference statements w.r.t these postulates, seeking for properties that could characterise their behaviour.

Interestingly, one of the results of our analysis corroborates a well-known shortcoming in a popularly used preference semantics. It also reveals certain semantics that have very interesting properties, regarding the composition/decomposition of preferences.

Chapter 4 Our final contribution is the design of a personalised decision support system using comparative preference statements. We address a single user for personalised decision support by eliciting their current preferences and providing a recommendation based exclusively on these preferences. Relying both on statistically-driven AI for polarised feature detection and logic-based AI gleaned from theoretical studies about reasoning with preferences, our system consists of (1) a preprocessing unit, (2) an interactive preference elicitation unit, (3) a preference logic based reasoning engine and (4) a final recommendation module which ensures that the computed recommendation list is satisfactory (i.e. resolves instances of empty/too large recommendation lists before providing final results). Our accompanying implementation is centred on the problem of choosing a hotel, based on an appropriate corpus of hotel reviews.

1
Literature Review

Introduction

Researchers have long been involved in the study of preferences for their utility in solving problems related to decision-making. With the advent of artificial intelligence in the field of computer science, this topic has gained particular interest within the AI community, and is one of the core issues in the design of any system that automates or supports decision-making. We therefore survey its salient points in this chapter, to place the contributions presented in this thesis within their scientific context.

The basic elements that constitute the handling of preferences in AI can be identified by asking the following questions: (1) What mathematical structures accurately describe the cognitive notion of preference, and which kinds are of particular interest in the AI context? (2) What forms of reasoning incorporate preferences in decision making, and how do we actually compute with them? (3) Are these mathematical structures easily described in practice? If not, what formalisms ease their transition from theory to practice? (4) Once a formalism is established, how can we obtain preferences from users, agents, etc. that comply with the formalism?

Answering the first two questions establishes the theoretical foundations of handling preferences, and will be described in §1.1. The next two questions address their passage from theory to practice and are respectively answered in §1.2 and §1.3. Having covered preference handling in AI, we complete our literature review in §1.4 with a survey of some of its current applications in personalised decision support. The advances presented in the last two sections will show how preferences in AI have exploded today into an exciting and crucial aspect of artificial intelligence.

1.1 Theoretical Foundations

The past two decades have seen the emergence and fruition of the field of "preferences" in AI, with several research groups, dedicated workshops, conferences and editorial endeavours aimed at promoting this multidisciplinary topic (Goldsmith and Junker, 2009, Brafman and Domshlak, 2009, Fürnkranz and Hüllermeier, 2010, Kaci, 2011, Domshlak et al., 2011, Pigozzi et al., 2015). We particularly mention the international multidisciplinary EURO working group for Advances in Preference Handling [1].

These studies find their roots outside the field of computer science in the distinct areas of economics, operations research and philosophy, with the formal developments of:

— decision theory, social choice and game theory (Von Neumann and Morgenstern, 1944, Arrow, 1953), and

— the logic of preferences (Halldén, 1957, Von Wright, 1963).

Building upon these works, AI researchers established the theoretical base for handling preferences:

1. preference models and their numerical representation and reasoning theories (Fishburn, 1970, Krantz et al., 1971, Roubens and Vincke, 1985, Fishburn, 1999), and
2. newer preference logics, enriching the former through the investigation and formalisation of non-monotonic reasoning (McDermott and Doyle, 1980, McCarthy, 1980, Shoham, 1987b, Kraus et al., 1990).

We shall look at these two aspects more closely in this section. AI researchers then constructed upon this base to develop the formalisms that allow putting preferences into practice: this will be the topic of the next section.

1.1.1 Modelling Preferences

Implicit in the word preference lies the idea of comparison. A preference exists only when alternatives can be compared and evaluated according to one's liking. Now, a first step in the scientific exploration of a concept, especially when it is abstract, is the building of its *model*, i.e. a mathematical structure that can capture the essential properties of the paradigm and thereby aid in the concretisation of the said concept. That being so, we can intuit a model for preferences in decision support to be an *ordering over a set of possible outcomes or alternatives*.

Formally, *preference modelling* requires mathematically defining a relation that compares outcomes and identifying the different properties that this relation

1. http://preferencehandling.free.fr/

could have, on the basis of which different preference models can be established. This is based on and adapted from existing notions in order theory.

We begin by reviewing these, and the basic mathematical principles involved in them. Having done so, we show how preference relations can be defined following the same principles and then make a survey of some current preference models. Throughout this section, we adopt the notation from Roubens and Vincke (1985).

Order Theory: Definitions. A binary relation R can be described as a pairwise comparison of elements in a set S which expresses their mutual R-relationship.

Definition 1 (Binary Relation)**.** Let $R \subseteq S \times S$. If $(a, b) \in R$, then one says that the element a is in binary relation R to the element b. An alternative notation for $(a, b) \in R$ is aRb.

The following are some basic properties of binary relations:
— R is *reflexive* iff $\forall a \in S$, aRa;
— R is *irreflexive* iff $\forall a \in S$, *not*(aRa);
— R is *complete* iff $\forall a, b \in S$, we have aRb or bRa;
— R is *transitive* iff $\forall a, b, c \in S$, if aRb and bRc then aRc;
— R is *symmetric* iff $\forall a, b \in S$, if aRb then bRa;
— R is *antisymmetric* iff $\forall a, b \in S$, if aRb and bRa then $a = b$;
— R is *asymmetric* iff $\forall a, b \in S$, we have *not*(aRb and bRa).

Depending on the properties satisfied by a given binary relation, it can be characterised as an order relation where the order could be: a *quasi-order* or *preorder*; a *partial order* or just an *order* by abuse of language; a *total order* or *linear order* or *complete order*. Sets equipped with order relations are known as *ordered sets*. These are formally defined as following:

Definition 2 (Ordered Sets)**.** Let S be a set. An *order* (or partial order) on S is defined as a binary relation $\leqslant$ on S which is reflexive, transitive and antisymmetric.

A set S equipped with an order relation $\leqslant$ is said to be a *partially ordered set*, also known by its shorthand notation *poset*.

When $\leqslant$ is not necessarily antisymmetric, it is defined as a *quasi-order*, or *preorder*.

When $\leqslant$ is complete, it is said to be a *total order*, and S is a *totally ordered set*, also known as a *chain*.

We now show how preference relations can be defined and consequently lead to different ordering structures, each being a preference model.

Preference Relations. Following the afore-mentioned definition of binary relations and their basic properties, we are now well-equipped for modelling preferences.

A *preference relation* $\succeq$ is defined on a set O of *objects/outcomes/alternatives* [2] that can be compared or evaluated according to their satisfaction of preference.

Definition 3 (Preference Relation). Let O be a set of outcomes, $\succeq$ a binary relation $\subseteq O \times O$ and $o, o' \in O$ such that o is in binary relation to o' w.r.t. $\succeq$. Then the notation $o \succeq o'$ is read as "o is at least as preferred as o'", and $\succeq$ is defined as a preference relation.

Defined in this way, $\succeq$ satisfies each of the properties of binary relations whenever their respective conditions are met. Furthermore, in the context of decision support, we distinguish between *strict*, *indifferent* and *incomparable* relations in the following way:

- when $o \succeq o'$ holds, but $o' \succeq o$ does not hold, then o is *strictly preferred to* o' and we write $o \succ o'$;
- when both $o \succeq o'$ and $o' \succeq o$ hold, then o is *indifferent to* o' and we write $o \approx o'$;
- when neither $o \succeq o'$ nor $o' \succeq o$ hold, then o is *incomparable to* o' and we write $o \sim o'$.

Thus, $\succ$ is asymmetric; $\approx$ and $\sim$ are symmetric; and if $\succeq$ is reflexive, then $\approx$ is reflexive and $\sim$ is irreflexive. These properties lead to ordering the set O of outcomes. These are known as *preference structures*, and correspond to the ordered sets seen above. For the purposes of this thesis, we recall the distinction between the notion of total/partial orders/preorders. For a complete list of the different possible structures, we refer the reader to (Öztürk et al., 2005, p.10).

Preference Structures. We have seen how the properties of different preference relations (e.g. $\succ, \sim, \approx$) can affect the ordering of a set of outcomes, based on the conditions they satisfy. Adding the hypothesis that each preference relation is uniquely characterised by its properties, the concept of *preference structures* is formalised as following:

Definition 4 (Preference Structure). A preference structure is a collection of preference relations defined on the set O such that: $\forall(o, o') \in O \times O$, at least one, and only one, relation is satisfied.

In particular, using the three types of preference relations we distinguished in our context, we say that:

Given $\succeq$, the triple $(\succ, \approx, \sim)$ is a preference structure induced by $\succeq$, and the properties of $\succeq$ are those of its associated relations in $(\succ, \approx, \sim)$. We discern the four following preference structures:

2. Henceforth we shall only use the term 'outcomes'.

partial preorder: $\succeq$ is reflexive and transitive as the associated $\succ$ is transitive, $\approx$ is reflexive and $\sim$ is not empty.

total preorder: $\succeq$ is reflexive, transitive and complete as the associated $\succ$ is transitive, $\approx$ is reflexive and $\sim$ is empty.

partial order: $\succeq$ is reflexive, transitive and antisymmetric as the associated $\succ$ is transitive, $\approx$ is reflexive and composed of (o, o) pairs only, and $\sim$ is not empty.

total order: $\succeq$ is reflexive, transitive, complete and antisymmetric as the associated $\succ$ is transitive, $\approx$ is reflexive and composed of (o, o) pairs only, and $\sim$ is empty.

Note that in the case of a total preorder, the $\approx$ relation being reflexive, transitive and symmetric, it is an equivalence relation. Let $E_1, ..., E_n$ be the set of equivalence classes induced by $\approx$. We have,

1. $\forall i = 1...n, \quad E_i \neq \emptyset$,
2. $E_1 \uplus \cdots \uplus E_n = O$,[3]
3. $\forall o, o' \in E_i, \quad o \approx o'$.

Consequently, we can say that we have an *ordered partition* on O given the total preorder $\succeq$, written as $(E_1, ..., E_n)$, if and only if the following condition holds:

$$\forall o, o' \in O, \quad o \succ o' \iff o \in E_i,\ o' \in E_j \text{ with } i < j.$$

When manipulating preference structures, we use the notion of *(un)dominated outcomes* or *maximally/minimally preferred outcomes*, and formally define this as:

Definition 5 (Set of Maximal (resp. Minimal) outcomes)**.** Let $\succeq$ be a partial preorder over a set of outcomes O. The set of the maximal (resp. minimal) outcomes of $O' \subseteq O$ w.r.t. $\succeq$ is written as $\max(O', \succeq)$ (resp. $\min(O', \succeq)$) and defined as $\{o | o \in O', \nexists o' \in O', o' \succ o\}$ (resp. $\{o | o \in O', \nexists o' \in O', o \succ o'\}$).

Now that we have explored the mathematical means of modelling preferences, we move on to understanding the theoretical construct of reasoning with these preference models.

1.1.2 Reasoning with Preferences: Principles and Mechanisms

Looking at the preference models and the properties they enjoy, by what mechanism would one be able to apply them to a given situation and manipulate them to obtain satisfactory results? If one looks at the problem from a purely

3. We use the notation $\uplus$ tfor *disjoint union* to express the union of two disjoint sets.

theoretical/philosophical standpoint, what principles does this involve, and are there distinct mechanisms that offer solutions in different situations? We term this problem that of reasoning with preferences, and look at these principles and mechanisms in the following part of this section.

Faced with the problem, our first intuition would be to fall back to the stable ground of manipulating real numbers, and to base ourselves on the similar ordering properties enjoyed by the set of real numbers. This was, indeed, the first recourse for economists and decision theorists, and we begin by describing the numerical mechanisms that map the set of outcomes to the set of real numbers.

We then cover another way of tackling the problem from the more abstract, philosophical side, which is to analyse the very reasoning process and to define a formal system in which preferences can be described and manipulated from a set of basic axioms, following reasoning principles in logic. These systems are known as preference logics.

Numerical Mechanisms The numerical representation of preference structures began in classical decision theory and decision analysis, by specifying a utility function $u : \Omega \to \mathbb{R}$ (Von Neumann and Morgenstern, 1944) which maps the set of outcomes to the set of real numbers. Formally, we have:

$$\forall \omega, \omega' \in \Omega, \omega \succeq \omega' \iff u(\omega) \geqslant u(\omega').$$

We can see that this function is therefore order-preserving. By virtue of this property, reasoning with preferences could be translated and performed numerically through mathematical analysis, using the representational theory of measurement (Krantz et al., 1971). This led to the development of several utility theories, as surveyed in Fishburn (1999), ranging from the simplest order-preserving function as shown above to more sophisticated ones where, for example, outcomes could be characterised not just by numbers or weights, but ranges of numbers, i.e intervals.

Logical Mechanisms Formalising the notion of reasoning with preferences, or developing *preference logics* began in philosophy, initiated by Halldén (1957) and championed by the now seminal work of Von Wright (1963). These first preference logics were based on classical logic, and provided a mechanism that interpreted preferences between propositional formulae.

Von Wright's system interprets preferences between two logical formulae Φ and Ψ using two technical devices: (1) the assumption that outcomes satisfy the formulae "everything else being equal", known as the *ceteris paribus proviso* and (2) the interpretation of Φ and Ψ more precisely as $\Phi \wedge \neg\Psi$ and $\Psi \wedge \neg\Phi$ respectively, using the logical connectives $\wedge$ and $\neg$, known as *Von Wright's expansion principle* (c.f. 3.1.1 on page 61 for a further discussion).

Simply put, the preference between Φ and Ψ is interpreted as "everything else being equal, I prefer an outcome satisfying Φ and not satisfying Ψ to an outcome satisfying Ψ and not satisfying Φ". For example, in "I prefer coffee to tea", 'coffee' and 'tea' would be interpreted as logical formulae and the statement would therefore read: "Everything else being equal, I prefer an outcome where I have coffee and no tea to an outcome where I have tea and no coffee."

Thus in formal terms, the system uses a reasoning mechanism that translates the preference between two logical formulae Φ and Ψ into that between two outcomes ω and ω' when:

1. ω satisfies $\Phi \wedge \neg\Psi$,
2. ω' satisfies $\Psi \wedge \neg\Phi$,
3. ω and ω' obey the *ceteris paribus proviso* w.r.t $\Phi \wedge \neg\Psi$ and $\Psi \wedge \neg\Phi$.

Building upon Von Wright's system, Van Benthem et al. (2009) extend and generalise it providing a complete, axiomatised logic that captures preferences with the ceteris paribus proviso.

Bienvenu et al. (2010) look at the existing preference logics from a computational angle and provide several complexity results, by addressing preference representation languages developed in AI research (we look at these in the following section). From this perspective, they provide a new logic making a trade-off between expressivity and complexity.

These mechanisms follow the classical property of logic known as the principle of *monotonicity of entailment*: "if a fact is derived on the basis of certain premises, then no additional premises can ever invalidate this fact". But what happens if additional premises can actually change this fact?

To answer that question, let us look at the following example of Tweety the bird, often taken up by researchers: "if Tweety is a bird, we can infer that Tweety flies; however, upon learning that Tweety is a penguin we retract our conclusion that Tweety flies (without having to question our assumption that most birds fly, or that Tweety is a bird).". This form of reasoning does not enjoy the property of monotonicity. It is known as the principle of *non-monotonic reasoning*.

Shoham (1987a,b, 1988) provided a unifying semantical framework which captures the idea behind all non-monotonic logics. This is the notion of selecting and working with the subset of valid models of a set of formulae, thereby favouring some interpretations over others. In his own words [1987b, p.389], "Non-monotonic logics are the result of associating with a standard logic a preference relation on models".

Investigating this principle in the context of reasoning with preferences led to an important addition to preference logic systems. Situations where preferences were uncertain, or needed to be revised based on subsequent information (e.g. defeasible preferences), could now be included and managed by defining newer mechanisms based on non-monotonic reasoning.

For example, let us suppose that in choosing a university for further studies, an individual would prefer a Paris university to a London university except if the university offers an optional drama course. Here, Paris universities are preferred to London ones in general, but this preference is reversed in the specific case where the university offers an optional drama course. These kinds of preferences are known as *defeasible preferences*. Simultaneously managing the specific reversed preference with the general preference requires the use of non-monotonic reasoning and can be done by applying the so-called the *specificity principle*, attributed to Yager (1983). We shall investigate this further in Chapter 3, §3.2.

1.2 From Theory to Practice: Preference Formalisms

The content of the previous section has given an indication to the body of work that has gone behind formalising preferences, allowing them to be exploited numerically and logically. For the former, the preference models we saw have a sound theoretical base, and their numerical counterparts, i.e. utility functions, can be directly used for preference-based reasoning.

On the more abstract side, preference logics have been defined and rigorously analysed to reason with the preference models. With this level of scrutiny used in their formalisation, we can get back to our original point of interest for preferences–their use in decision making–and explore the possibilities this opens up for decision support.

Seen in the simplest way, making a decision requires considering all possible scenarios before choosing the more favourable ones. Using preferences, this boils down to ordering the different scenarios according to one's preferences and then choosing from the better ranked ones. Now, theoretical research on preferences provides us with further details about the task: we can say exactly what properties our preference order enjoys and what kind of reasoning to use for inferring our preference relations. Thus, in providing decision support for agents (whether human or artificial), we can incorporate this fine grained knowledge to help them in their task. There is only one tiny hitch: all of this rests on a cognitive (of a human agent) and computational (of an artificial agent) capacity to describe a preference model.

With the exploding quantities of data available at our fingertips, it seems this is a cognitive and computational overload. Faced with a large number of alternatives, our natural way of expressing our preferences is to resort to an abstraction of the problem by describing piecewise information about our preferences. For example, "I like London more than Paris" is a partial description that can come in handy when a user is faced with the task of choosing, say, a university for further studies, and is confused about the one they would like to be in the most. These questions were tackled by researchers in decision theory and led to the development of the field of *qualitative decision theory*. As argued by Doyle and Thomason (1999, p.58),

> "The need for an approach to decision making based on partial information provides a powerful motivation for a qualitative decision theory. Common experience shows people offering partial, abstract, generic, tentative, and uncertain information in explaining their decisions. This information includes qualitative probability ("I'm likely to need $50,000 a year for retirement"), generic preference information ("I prefer investments in companies that respect the environment"), and generic goals ("I want to retire young enough to enjoy it"). If we wish a more direct model of the way people seem to think about decisions, we need to deal with such information and need models of deliberative reasoning that can make use of it in formulating and making decisions."

Researchers in AI have taken up these issues and have developed *compact preference representation languages* to represent such partial descriptions of user preferences. The preferences expressed in such languages are called *preference statements*, and use different means of completion to compute a preference relation (the preference model).

In this way, we now have a practically more suitable way to represent the mathematical model of preferences: using such languages, one can make the passage from a set of preference statements to a preference relation that is compactly represented by this set.

Today there exists a whole gamut of these languages, which have built on the solid foundational work of the past 60 years. This has led to the development of distinct *preference formalisms*, each formalism being a language to represent compact preferences, coupled with its method of computing a preference relation.

All preference formalisms use the same basic formal language and follow a general scheme to describe preference statements and therefore the preference models. We begin by describing this general scheme, then look at some well-known formalisms.

1.2.1 The General Scheme

The general scheme of compactly describing a preference model is based on (1) setting the context in which a preference relation can be defined, (2) describing the means of constructing a preference statement and (3) establishing the basis for how outcomes satisfy these preference statements.

The Context Let us consider the following components of a formal language:

— A finite set Σ of *variables* (denoted by upper-case letters) which describe the attributes or characteristics of an outcome,
— A finite set called the *domain*, $\mathrm{Dom}(X)$, for each variable X in Σ (denoted by lower-case letters) which contains all the values that can be assigned to that variable X,
— An *outcome* ω, which is the result of assigning a value to each variable X in Σ,
— The set of *all outcomes* Ω (i.e. the cartesian product of all variables in Σ).

These establish the context for all preference formalisms. A preference relation is defined on the set Ω, and evaluates the pairwise comparison of outcomes $\omega, \omega' \in \Omega$, based on their satisfaction of the given set of preference statements.

Preference Statement Construction Let $\mathcal{L}$ be a language based on Σ. Mathematical formulae (numerical or logical) built in $\mathcal{L}$ lead to the construction of preference statements. Thus, functions such as utility functions represent numerical *preference statements* constructed in $\mathcal{L}$.

For logical preference statements, by abuse of language, we can say that for each variable $X \in \Sigma$, we have the corresponding logical proposition X. Thus logical formulae are built by combining such propositions using the standard logical connectors $\wedge$, $\vee$ and $\neg$, and preference statements express the preference between two logical formulae p and q.

We say that a given preference statement "p preferred to q" can be interpreted as a preference over outcomes "ω preferred to ω'" when ω *satisfies* p and ω' *satisfies* q, where the satisfaction is formalised as following:

Definition 6 (Preference Satisfaction in $\mathcal{L}$)**.**

— $\mathrm{Mod}(p)$ is defined as the set of outcomes that make the formula p true (also called the set of p-outcomes),
— we say that an outcome ω *satisfies* p when we have $\omega \in \mathrm{Mod}(p)$,
— conversely, we say that ω *does not satisfy* p when we have $\omega \notin \mathrm{Mod}(p)$.

1.2.2 Compact Preference Languages

In this section, we visit the languages that form part of some well-known preference formalisms in the literature.

1.2.2.1 Logical Languages

All the compact preference languages in this category express preferences with the use of logical formulae, and order the outcomes by selecting maximal ones according to their satisfaction of these formulae.

These languages further fall into two sub-categories: weighted logics and conditional logics. The use of different kinds of logics brings about various solutions to the task at hand, each one suited to a particular kind of problem.

Weighted Logics Logical languages in this category all pivot around one common element: a weight that characterises the importance of a formula. This weight could be associated to:

- the aggregate penalty of every preference [4] violated as in *Penalty Logic* (Haddawy and Hanks, 1992),
- the penalty of violating preferences (i.e. the higher the weight, the greater the penalty of violating the preference, more important the preference) as in *Possibilistic Logic* (Dubois et al., 1994), or
- the reward of satisfying preferences (i.e. the higher the weight, the greater the reward of satisfying the preference, more important the preference) as in *Guaranteed Possibilistic Logic* (Benferhat et al., 2002b) and *Qualitative Choice Logic* (Brewka et al., 2004).

These characterisations are indicative of the *priority* associated with each preference statement, and often produce varying preference relations because of the different aspects they emphasise. For a detailed description of the current landscape of weighed logics in AI literature, we invite the reader to consult the dedicated special issue (Dubois et al., 2014).

Conditional Logics In contrast to weighted logics, conditional logics are based on a comparative evaluation of formulae. More explicitly, in conditional logics, preferences are defined as *comparative preference statements* such as "I prefer comedies to action movies". The inherent preference of "comedies" to "action movies" is interpreted as a preference of the form "comedies and not action movies" over "action movies and not comedies". This distinction is made to prevent occurrences where movies could be both "comedies" *and* "action movies", in which case the preference would make no sense. This is known as the *von Wright expansion principle*.

Logically, these statements are represented as $p > q$ where $p \wedge \neg q$ and $q \wedge \neg p$ are the formulae to be compared. An outcome satisfies $p > q$ iff it satisfies $p \wedge \neg q$. Following the conditions of preference satisfaction (Def. 6), this can be interpreted as $p \wedge \neg q$-*outcomes preferred to* $q \wedge \neg p$-*outcomes*.

Thus, these statements can be interpreted as $\mathrm{Mod}(p \wedge \neg q)$ preferred to $\mathrm{Mod}(\neg p \wedge q)$. Translating this preference over sets into the pairwise comparison of individual outcomes presents several possibilities, depending on how rigorously each outcome in both sets is required to satisfy the preference. Taking up our example of choosing a university, "I like London more than Paris" could

4. represented as a logical formula

either impose that *all* London universities are preferred to *all* Paris universities, or loosen the requirements and allow exceptions to the preference.

Different semantics have been proposed in the literature to account for this, some of which are:

Strong Semantics (Benferhat and Kaci, 2001, Wilson, 2004) where any outcome satisfying $p > q$ is preferred to any outcome satisfying $q > p$,

Ceteris Paribus Semantics (Von Wright, 1963, Doyle and Wellman, 1994, Hansson, 1996) where any outcome satisfying $p > q$ is preferred to any outcome satisfying $q > p$ if the two outcomes have the same valuation over variables not appearing in p and q [5],

Optimistic Semantics (Pearl, 1990) where any one of the maximal (see Def. 5 on page 11) outcomes satisfying $p > q$ is preferred to any one of the maximal outcomes satisfying $q > p$,

Pessimistic Semantics (Benferhat et al., 2002b) where any one of the minimal (see Def. 5 on page 11) outcomes satisfying $p > q$ is preferred to any one of the minimal outcomes satisfying $q > p$,

Opportunistic Semantics (Van der Torre and Weydert, 2001) where any one of the maximal outcomes satisfying $p > q$ is preferred to any one of the minimal outcomes satisfying $q > p$.

We now move to the final set of preference representation languages, the graphical languages.

1.2.2.2 Graphical Languages

Preferences are also represented graphically, so as to highlight their inherent structure, mainly in terms of their dependencies. Following are some of the well-known languages:

— *Generalised Additive Independence Networks (GAI-nets)* (Gonzales and Perny, 2004) are a visual representation of dependencies in utility functions,
— *Conditional Preference Networks (CP-nets)* (Boutilier et al., 1999, 2004) are based on conditional logic, namely ceteris paribus semantics, and represent the dependencies between variables,
— *CP-nets with Utilities (UCP-nets)* (Boutilier et al., 2001) combine the two concepts by extending CP-nets with utilities.
— *CP-nets with Tradeoffs (TCP-nets)* (Brafman and Domshlak, 2002) extend CP-nets with information about the relative importance between variables,
— *Conditional Importance Networks (CI-nets)* (Bouveret et al., 2009) extend the notion of the importance between variables to that between sets of variables, ceteris paribus.

5. This is a commonly used interpretation of the semantics.

The algorithms that these formalisms use to compute preference relations exploit their graphical structure using *independence-based methodology*. Here the graphical framework itself constitutes an instance of direct logical reasoning about the preference expression.

1.2.3 Bipolar Preferences

When humans reason in a decision-making process, they often have a predisposition to simultaneously consider positive and negative affects. The explicit handling of these two opposing effects is known as bipolarity.

The interrelations between what we call positive and negative are quite complex; as aptly put by Dubois and Prade (2008, p.867):

> "There are several forms of bipolarity according to the nature and the strength of the link between the positive and the negative aspects; in the most constrained form, the positive is just the mirror image of the negative and they are mutually exclusive. A looser form of bipolarity considers a possible coexistence between positive and negative evaluations, while a duality relation between them is maintained. In the loosest form, the positive and the negative sides express pieces of information of a different nature."

In decision-making, bipolarity concerns the opposition between constraints and goals. We say that,

— negative preferences are constraints that state which solutions to a problem are unfeasible: accumulating these leads to a decrease in possible solutions;

— positive preferences state desires/goals which should be satisfied as best as possible for solutions: accumulating these leads to an increase in possible solutions.

Researchers have looked into representing this bipolar aspect using the languages we saw in this section, and have revealed some interesting results that could be useful in implementing methods in decision-support. We now look at two such examples of how this can be done in particular for logical languages.

In Weighted Logics Let us consider two of the weighted logics we described in this section: *possibilistic logic* and *guaranteed possibilistic logic*.

The former uses a weight to characterise the penalty of violating preferences, thus inducing a *tolerance distribution* on outcomes, and the latter uses a weight to characterise the reward of satisfying preferences, thus inducing a *satisfaction distribution* on outcomes.

Benferhat et al. (2002b) represent negative (resp. positive) preferences using weights which quantify the *rejection* (resp. the *minimal guaranteed satisfaction*) of violated (resp. satisfied) preference formulae. In this way, they show that a set of negative (resp. positive) preferences can be encoded using a possibilistic (resp. guaranteed possibilistic) logic base, thereby inducing a tolerance (resp. satisfaction) distribution.

Analysing further w.r.t the addition of preferences, these distributions exhibit a dual nature, revealing bipolarity in weighted logics. Considering additional preferences for a given set may only:

— *decrease* the tolerance in possibilistic logic, and
— *increase* the satisfaction in guaranteed possibilistic logic.

This corroborates the respective definitions of negative and positive preferences. Thus, bipolarity can be conjointly expressed in weighted logics using possibilistic logic and guaranteed possibilistic logic representations.

In Conditional Logics Kaci (2012b) addresses the bipolar nature of information, and resolves it in the specific case of comparative preference representation.

She shows that the non-monotonic principle of maximal and minimal specificities can be used to define aggregator functions which allow interpreting comparative statements quantitatively. Making an analogy with the distributions we saw for weighted logics, she shows how *optimistic (resp. pessimistic) semantics* reason in the same way as negative (resp. positive) preferences.

She then provides a set of postulates proposed in the literature describing the different situations that one may encounter, and a representation theorem which characterises subsets of these postulates according to their respective satisfaction of negative and positive information. This reveals the dual nature of optimistic and pessimistic semantics, and corroborates their ability to model negative and positive information respectively.

With this brief overview of the different existing preference formalisms [6], we now address the task of acquiring preferences. In the context of decision support systems, this task allows the system to interpret user preferences.

1.3 Preference Acquisition

By way of the different languages surveyed in the previous section, we saw how preferences could be modelled using compact representations to provide preference formalisms that enable decision-support in real-world applications.

6. For details and an in-depth comparative analysis, see Kaci (2011, Ch.3,4).

Given a particular decision-making scenario, the final step in implementing these formalisms is acquiring the agent's preferences.

There are several ways in which this can be achieved, and two distinct branches have emerged over the years, each focussing on a different aspect of acquiring preferences. *Preference elicitation* methods are process-oriented, whereby user interaction leads to the construction of preference formalisms; while *preference learning* is data-oriented, and applies machine learning techniques on available data to predict a model following a specific preference formalism.

For example, let us say a school counsellor must provide university recommendations for each of this year's outgoing students. If they were to use preference elicitation, they would adopt a method of questioning each student to establish their preferences. On the other hand, making predictions about student preferences using machine learning algorithms on available data about each student (e.g. progress reports, marks, courses followed, etc.) would fall into the category of preference learning.

We shall see in the course of this section that advances in these two fields have been depth-first rather than breath-first, which has progressively caused a bottleneck in preference acquisition: when a breakthrough has been made, subsequent contributions have flowed over the years, while previously unexplored directions remain untapped for their merits.

1.3.1 Preference Learning

One of the two main methods used in artificial intelligence for acquiring preferences is *preference learning*. It is a specialised method of machine learning, where the system learns from users' past preferences to make predictions about unseen user preferences. Recommender systems in online shopping websites often employ such techniques for personalised shopping recommendations. The first comprehensive book covering the entire topic Fürnkranz and Hüllermeier (2010) includes, in particular, a systematic categorisation according to learning task and learning technique, and furnishes a unified notation for preference learning.

Amongst the preference formalisms visited in the previous section, learning methods have been developed for (1) utility functions and (2) pairwise preference relations, which involve conventional methods from machine learning such as classification and regression [7], and (3) CP-nets from examples using different machine learning approaches as can be found in the works of Lang and Mengin (2009), Koriche and Zanuttini (2010), Liu et al. (2013).

7. references can be found in Fürnkranz and Hüllermeier (2010)

We mention very few references here, and only those that pertain to the formalisms visited in this chapter. The interested reader will find the book mentioned here to be extremely instructive.

1.3.2 Preference Elicitation

The other primary method in preference acquisition is *preference elicitation* where preferences are the result of interactive processes with the user. These can further be categorised under (1) numerical utility elicitation and (2) qualitative preference elicitation.

Utility Elicitation. Numerous methods have been designed for this kind of elicitation, based upon querying the user about the relative importance of every possible outcome in terms of each decision criterion. The elicitation of utility functions for multi-attribute and multi-criteria settings goes back to the work of Keeney and Raiffa (1976) and recent works have been surveyed by Braziunas and Boutilier (2009), Viappiani (2014). Amongst the preference formalisms visited in the previous section, those that have been well adapted to this type of elicitation are the GAI models (Gonzales and Perny, 2004, Braziunas and Boutilier, 2007).

The main difficulty tackled in these methods is exponential outcome spaces, defined by the possible values of outcome attributes. These occur when decision problems are defined by multiple attributes. Taking the example from Braziunas and Boutilier (2009), if we consider a sophisticated flight selection, then possible outcomes are defined by attributes such as trip cost, departure time, return time, airline, etc. and their number is therefore exponential. Utility elicitation methods differ in the elicitation strategies they devise and the optimisation of outcomes they provide, which result in reducing the number of queries made to the user.

Qualitative Elicitation The need for qualitative elicitation over the numerical elicitation of a utility function can often be simpler for the user involved in the elicitation process, and reasons for this can be found in the early discussions promoting qualitative decision making. As remarked by Doyle and Thomason (1999), "Traditional decision theory provides an account of the information that, in principle, suffices for making a rational decision. In practice, however, the decision maker might have never considered the type of choice in question and so might not happen to possess this information."

Using this stream of thought, Domshlak (2008) introduces a general scheme for qualitative preference elicitation as a means of addressing the bottleneck of preference acquisition for the different preference formalisms available, and specifies how it could be applied to the CP-net formalism. Guerin et al. (2013) push this application further and provide algorithms for eliciting user preferences that can be represented as CP-nets. It is interesting to note here that although the method described is an elicitation process, it bases itself on the CP-net learning techniques we saw in §1.3.1 to refine its results.

We also find other interactive methods that fall in this category in the surveys provided by Peintner et al. (2009), Pu et al. (2012), detailing the complex procedures used in each one. The tools surveyed focus on acquiring preferences through the user's behaviour, handling conflicting preferences and revising initial preferences. They involve techniques such as *example critiquing*, where the users are prompted to choose their preferred outcomes from a selection of examples presented to them. These methods differ from each other in terms of the criteria used: (1) feature-oriented systems describe preferences based on product features, while (2) needs-oriented systems describe preferences based on the users' personal needs.

User-Based Preference Elicitation In all the elicitation methods surveyed until now, the user is solicited to improve the quality of the formalism adopted by the system. Since the focus is on the formalism, this may not necessarily correspond to the user's own model of preferences. Thus the information provided by the user is *interpreted* as preferences by the system, as opposed to *expressed* in a natural way by the user. For instance, this information is a numerical input (as in the case of numerical elicitation) or a visual input (example critiquing) or a simple yes-no query response about pairwise attribute comparisons (for CP-net elicitation); and further information can be required to resolve the trade-offs that might crop up.

Now, if we consider the elicitation problem from the user's point of view, there are two primary considerations that have been overlooked: (1) the linguistic expression of preferences and (2) the role of preferences in the psychological construct of decision making. These shed light on the way humans make their decisions, which decision support systems could benefit from. There have been contributions which focus on the linguistic expression of preferences (Cadilhac et al., 2012, Nunes et al., 2015) and those that use heuristics from psychology (Nunes et al., 2015, Allen et al., 2015).

Cadilhac et al. (2012) develop a dialogue annotating method using segmented discourse representation theory, identifying preferences by discovering linguistic terms for outcomes and their dependencies. The natural language processing (NLP) techniques employed focus on extracting outcomes, and preferences are represented as a result of detecting how these extracted outcomes are ordered.

Nunes et al. (2015), Allen et al. (2015) describe the design of an entire decision support system. This means that they tackle the problem of acquiring preferences, modelling them in a specific formalism and reasoning with them to generate recommendations. This being the topic of our next section, we mention them here for their pertinence to preference elicitation and shall discuss them further in the next section.

1.4 Preferences in Decision Support

As evidenced by the contributions studied in this chapter, the fields of qualitative decision theories and preferences in AI have frequently crossed paths, and there is a palpable optimism in their collective aim of converging and breaking ground together. It was not so long ago that Domshlak (2008) mused on a "chicken-and-egg" paradoxical deadlock system and said:

> "On the one hand, it is only natural to assume that reasoning about user's preference expressions is useful in many applicative domains (e.g., in online catalog systems). On the other hand, to our knowledge, no application these days allows its users to express any but trivial (e.g., "bag-of-word") preference expressions. It seems that the real-world players wait for the research community to come up with a concrete suggestion on how natural-language style preference expressions should be treated, while the research community waits for the real-world to provide it with the data essential to make the former decision. It is clear that this deadlock situation should somehow be resolved, and we believe that now this should be a primary goal for both sides."

Since then, there has been an explosion of advances on the practical, technical front in various disciplines of AI with the advent of Big Data and the flourishing of machine learning and natural language processing techniques. There is clearly 'a new hope' since Domshlak's musings.

For decision-support in particular, this can be seen through the work of Chen and Pu, who provide regular surveys and in-depth analysis of preference handling methods, revealing pitfalls and providing guidelines for future research. They began by surveying existing methods in preference elicitation in 2004–the early years of the development of preference formalisms in AI–and have since seen the growth and development of real-industry applications such as preference-based recommender systems in 2009, 2010a, 2010b, 2012.

Contributions of preferences in decision support continue to flow in, with advances in (1) existing decision-support systems such as recommender systems and (2) stand-alone decision-support systems which could potentially be reused for other purposes. The latter category generally comprises of contributions that have made a significant breakthrough, opening up further avenues for implementations. We now present a few among these which focus in particular on handling preferences as seen in this chapter.

1.4.1 Preferences in Recommender Systems

In this day and age when information search and selection is increasingly performed online, recommender systems have proven to be a valuable way for online users to cope with information overload. They have become popular tools in electronic commerce, and are found in practically every virtual interaction we have, be it shopping, listening to music, watching videos or reading the news.

These systems use algorithms that approximate, or predict, possible recommendations on *items*, based on available information about the users and/or the items in question. They generate *recommendation sets* on the basis of *similarity measures* to compute similar users and/or similar items or *predicted ratings* on items. Computing such recommendations has become a veritable testing ground for artificial intelligence (AI) technologies with their range of applicability and diversity of approaches. The growing number of users, content and social media have provided a fertile ground for the improvement of current approaches, and the state of the art today performs remarkable feats of artificial intelligence.

Traditional approaches to recommendation are grounded on well-proven technology such as *collaborative filtering, machine learning, content analysis,* etc Resnick and Varian (1997), Kantor et al. (2011). They exploit the increasing amount of data available, and provide recommendations about items using

— user-based (preferences of similar users),
— content-based (preferences about similar features),
— knowledge-based (to reason about items which respond to user requirements), or
— hybrid

methods Adomavicius and Tuzhilin (2005), Balabanović and Shoham (1997), Burke (2002), Burke and Trewin (2000), Chen et al. (2015).

User- and content-based approaches stem from the information retrieval (IR) community, and are *quantitative*, as they depend on and manipulate data that is quantified as weights (for user preferences, ratings), or similarity measures (for users and content). The former bases itself on the similarity of different users' profiles to predict a match; while the latter uses techniques to extract features from the content of products (e.g. text, music, etc.) and weigh them against those of user profiles to find a match. Different systems based on the same approach will differ in the machine learning techniques applied and/or the way they are applied.

Knowledge-based approaches can use *qualitative* information obtained from users to guide them through the decision process. The former are more efficient but only provide predictions, while the latter adhere closely to the decision problem and provide exact solutions, but at the cost of complex algorithms.

Hybrid approaches, initiated by Balabanović and Shoham (1997), successfully combine the advantages of each. Gleaning from the overviews by Burke (2002), Adomavicius and Tuzhilin (2005), Koren (2010), Koren et al. (2009), Lu et al. (2015), newer approaches can be identified, based on the availability of newer data. For example, the advent of GPS positioning in mobile phones has led to the demographic approach to recommendations. We now look at the pertinence of this increasingly popular application of decision-support to the topics covered in this chapter.

Preference-based Recommender Systems The study of preferences in AI has been used for decision support systems such as preference-based web applications e.g. product search, recommender systems, personal assistant agents, and personalised user interfaces, stressing upon the growing importance of user-involved preference acquisition and recommendation, as initiated by Chen and Pu.

Preference learning has been well-suited to existing recommendation algorithms as it uses techniques in machine learning to learn and predict preference models from data describing the user's behaviour or past preferences Fürnkranz and Hüllermeier (2010), Liu et al. (2015). These preference-based approaches usually describe the items in terms of their *features*, or attributes, and determine or predict the users' preferences about these features.

CP-nets have successfully been implemented in recommender systems using this approach. Liu et al. (2015) build upon these and make a technical breakthrough by tackling the space complexity of representing them and the computational complexity of their learning models. They prove that the expressive ability of the quadratic polynomial is stronger than that of the linear function when approximating conditional preference, and use the former in a matrix-factorisation rating-based recommender system to confirm its superiority.

Preference elicitation has been shown to be an important phase for decision-support, since preference learning-based systems cannot provide recommendations about users' current preferences, or if their preferences themselves have evolved. There is a recent trend in evaluating this approach from the user's perspective to bring improvements to it (Parra and Brusilovsky, 2015, Ekstrand and Kluver, 2015, Chen et al., 2015, Chen and Pu, 2014, Das and Morales, 2013, Konstan and Riedl, 2012, Pu et al., 2012).

For example, Pommeranz et al. (2012), Knijnenburg et al. (2012) address issues raised by Pu et al. (2012) and design experimental studies testing the guidelines proposed, based on eliciting preferences specifically adapted to the algorithms used in recommendation systems. Their results agree with and complement each other, which shows the relevance and need for a user-centric evaluation of recommender systems. The interesting aspect of their studies is to expose the user to the inner workings of recommender systems (i.e. the kind of data involved in the algorithms) and to analyse their response to this. This not only reveals how the user intuitively interacts with such data, but also how they feel about the pertinence of the very concept of recommender systems. The significant contribution of such studies is that they confirm the fact that the quality of recommender systems does not depend solely on the accuracy of its predictions, but also on user-experience in general, and that much can be gleaned from user-response to improve the algorithm design in recommender systems.

Incorporating an explicit elicitation process, *preference-based product ranking* approaches have been used in recommender systems. Items are represented by a set of features and the user's preference is elicited in the form of weights or value criteria Smyth (2007), Musat et al. (2013), Poriya et al. (2014). The formulae used are based on the research developed in multi-attribute utility theory (MAUT) (Keeney and Raiffa, 1976).

Conversational recommender systems (Ricci et al., 2006, Viappiani et al., 2006, McSherry, 2005, Bridge et al., 2005) also find their place in this category, as they guide the user through questions, suggestions and explanations. This approach is a promising way of incorporating AI theories for knowledge-based recommendation. A notable example is the use of comparative preference theories (Wilson, 2009) in such a system (Trabelsi et al., 2011).

The use of preferences in these recommendation approaches adds a personalised dimension to the prediction algorithms in user- and content-based recommender systems. Their use in knowledge-based recommender systems, however, remains a less-explored direction. The results shown in Trabelsi et al. (2011) are promising w.r.t the inclusion of comparative preference theories for information recommendation, as they point to the pertinence of exploiting the expressive nuances of the form of preference representation.

We believe the trend of improving preference elicitation techniques will bridge the gap between the fields of recommender systems, qualitative decision theories and preferences in AI, allowing many theories that have stayed abstract to finally bear fruit in our everyday lives. With the very recent trend of developing interactive AI bots to provide a variety of services to users, these approaches will have a handy platform for pointed applications. For now, there are stand-alone systems that achieve this goal experimentally, and we describe a few among these in the remaining part of this section.

1.4.2 Preferences in Stand-Alone Decision-Support Systems

Alanazi et al. (2012) propose an interactive online shopping system, addressing the need to elicit user preferences to provide recommendations. They include both hard constraints and preferences in their system, implementing the former under the Constraint Satisfaction Problem (CSP) framework and the latter according to the CP-net formalism. Their use of a previously existing algorithm to approximate CP-nets to a Soft Constraint Satisfaction Problem (SCSP) allows them to (1) overcome the computational complexity associated with CP-nets and (2) combine hard and soft constraints in a weighted-CSP. They finally use the branch and bound algorithm to discover the best solutions for recommendation. In terms of the elicitation process, they restrict themselves to a given decision scenario with fixed variables and corresponding domains. The user then has the option of expressing their preferences qualitatively as conditional preferences or quantitatively as hard constraints from drop-down menus for each variable.

Nunes et al. (2015) use both the linguistic and psychological considerations in their contribution, and design their system based on the way humans use trade-offs and resolve conflicting preferences to make their decisions. Their approach is to (1) allow the user to describe what they call high-level preferences (which are close to preferences expressed in natural language), (2) provide them with a recommendation with associated explanations (this is where they use heuristics from psychology), so that they may in turn (3) refine their preferences. They finally generate a partially ordered set of recommendations organised into four levels: (1) the chosen option, (2) acceptable options, (3) eliminated options and (4) dominated options.

Allen et al. (2015) present the design of an experiment which highlights some of the significant computational, conceptual, ethical, mathematical, psychological, and statistical hurdles to testing whether decision makers' preferences are consistent with a particular mathematical model of preferences, using the test-case of the CP-net formalism in a human subjects experiment. The authors form a multi-disciplinary group of researchers, some eminent proponents of their respective fields, and advocate the collaboration between AI and psychology researchers as a mutually beneficial endeavour. We conclude this survey with a few amongst the many considerations from various disciplines they raise, since they are pertinent for the experiment design in our own contributions presented in this thesis:

1. User Uncertainty: The decision-maker's uncertainty in what to choose when faced with multi-attribute options in which attributes trade off in complex ways. To deal with this, they sketch the conceptual and mathematical challenges of defining uncertain choices induced by theoretical preferences that form CP-nets, using probabilities. They call these "probabilistic specifications". Addressing two major classes of probabilistic choice models, they look at the CP-net construct w.r.t each class, define a mathematical formula for each of the probabilistic specifications, and discuss the complexity of characterising them using mathematical structures.

2. Data bias: Statistical inferences from finite sample data require repeated observations either from multiple people or from a given participant. Eliminating potential biases and the effect of irrelevant variables must be tailored into the experiment design by implementing a variety of "cross-balancing" precautions such as distributing a given choice option in different locations to avoid attentional bias, and making the different cognitive tasks "equally complex" to balance cognitive load.
3. Correlation vs. causality: This has important implications for experimental methods as opposed to data mining or other approaches, since values in one variable may "cause" outcomes in another variable.
4. Falsifiability, diagnosticity and parsimony: The authors argue that since theoretical predictions motivate the experimental design, they precede data collection. Thus, one must be aware of some common errors in statistical inference: (1) supporting a theoretical claim by rejecting the null hypothesis of "no effect", (2) formulating mathematical models and inferring parameter values for the model using statistical methods, which brings their replicability into question, and (3) using Bayesian methods to compare different theories which vary only in their parsimony by weighing prior beliefs with empirical evidence, and penalising flexible models; this is a valid evaluation method, one must be careful not to draw scientific conclusions from very slight statistical effects.

Conclusion

In the course of this chapter, we followed the evolution of preferences in AI, offering a bird's eye view on the treatment of preferences from theory to practice, with particular emphasis on their role in the recent advances in decision support. In this context, the reader would have observed that preferences in AI have jumped onto the online recommendation bandwagon at the opportune moment, opening up a whole new field of practical applications.

Due to existing preference learning and elicitation methods that favour using numerical (e.g. utility functions) and graphical preference formalisms (e.g. CP-nets, GAI-nets), these have been used in several decision support systems. As regards CP-nets, we mention Allen (2015), who claims that the study of CP-nets has not advanced sufficiently for their widespread use in complex, real-world applications, and addresses these issues in his ongoing PhD research.

However, recalling the compact preference languages visited in §1.2, one can see that there remain efficient preference formalisms that have not yet seen the light of day in such systems, especially considering the logical languages. This is a commonly known bottleneck in preference acquisition. Our first contribution addresses this issue. We propose a method of acquiring user preferences expressed in natural language (NL) which could be adapted to comparative preference statements, to identify linguistic markers that reflect the different semantics (i.e. strong, optimistic, pessimistic, etc.) that could be associated with these comparative preference statements.

Our second contribution takes a deeper look into these semantics, by analysing their behavioural aspects using the postulates studied in preference logics and non-monotonic reasoning, which we briefly visited in §1.1.2. This allows us to formalise the intuition behind them, and gives us greater control on how to implement them in a decision-support environment.

Our final contribution is the design of a personalised decision support system which controls the entire pipeline from preference acquisition to recommendation. Due to the recently revealed inadequacies of comparing competing decision-support systems on the same over-used database, we follow the trend of evaluating our system by bringing humans into the loop right from the theoretical conception down to the experimental design. We note that due to the large-scale requirements of deploying such experiments as human-subject experiments, certain concessions have had to be made for the purpose of our evaluation: in ideal conditions, these would not be necessary.

2

Preference Acquisition: A Linguistic Analysis to Elicit Comparative Preference Statements from Natural Language Expressions

Introduction

The literature review presented in the last chapter revealed the extent to which research on preferences in artificial intelligence has contributed to personalised decision support systems. We identified several branches of the research that have successfully been incorporated into intelligent services such as recommender systems. Having made a broad review of existing preference representation and reasoning methods, we also pointed out those that have not yet seen the light of day in such systems due to a commonly known bottleneck in preference acquisition.

Delving a little deeper into the problem, we observed that it arises primarily for languages that represent preferences qualitatively. Ironically enough, these were developed precisely because they would respond more easily to the way we express our preferences! Our approach to alleviate this bottleneck is, therefore, to consider preference acquisition from the user's point of view, rather than the knowledge representation aspect it caters to.

Since the most instinctive and exhaustive manner of expressing human preferences is through natural language (NL), would a linguistic analysis of user preference expressions hold the clues to resolving this bottleneck?

Background

Our primary research objective is the acquisition of preferences that can subsequently be formally represented using a preference language in a personalised decision support system. We now recall those elements treated in Chapter 1 that are pertinent to this objective, and describe how we propose to reach it.

Preference Acquisition

Preferences in artificial intelligence can be acquired using two basic techniques: learning and elicitation. Preference learning covers methods using techniques in machine learning to learn and predict preference models from data describing the user's behaviour or past preferences. Preference elicitation is performed through an interactive process with the user.

Preference learning is more legitimate when the user is not a new customer and does not vary in their choices, or if it is assumed that individual behaviour imitates a computed average behaviour. It is, indeed, an important aspect of several recommender systems today.

Looking at acquisition from the user's point of view, psychologists assert that preferences are constructed, and not simply revealed, during the elicitation process (Slovic, 1995). That is, users do not know their preferences prior to the elicitation process. At a given point of time, a user may also have additional preferences, previously unknown to the system, or those that are different, even contradictory, to those they had before. Preference elicitation allows for these situations.

Comparative Preference Statements

Among the qualitative preference representation languages we have seen, comparative preference statements are intuitively similar to the NL expression of preferences: they are based on a comparative evaluation of logical formulae, and can be read for example as "I prefer comedies to action movies", or "if white wine is served, I prefer fish to meat". They belong to the sub-category called compact preference languages. The term 'compact' refers to their treatment of preferences as partial descriptions, and the use of different means of completion to compute a preference relation on the set of all outcomes Ω, that can be induced by a set of preference statements.

Formally, they are denoted $\alpha \rhd \beta$, and express the preference of α-outcomes over β-outcomes [1]. This preference is a partial description, since it only pertains to α- and β-outcomes. As a means of completion, these statements are accompanied with a semantics to interpret them, to compute a preference relation (partial or total preorder) over the entire set of outcomes, Ω, that is induced by a set of comparative preference statements.

1. or any subset of α- (resp. β-) outcomes.

For the statement $\alpha \triangleright \beta$, the accompanying semantics defines the way in which the set of α-outcomes is compared to the set of β-outcomes. Several semantics have been defined in the literature, as seen in Chapter 1. Based on the requirements of each of the semantics, this ordering may vary considerably.

Considering "prefer α to β", strong semantics imposes the most requirements, as all α-outcomes must be ordered above all β-outcomes. For example, in choosing a university for further studies according to the variables 'rank' and 'location', a strong preference for 'London' over 'Paris' would result in ordering all London universities above Paris universities, regardless of their rank. An NL form of the statement could be "*I definitely want to be in London*" and is not necessarily explicitly comparative. The important element here is the emphasising role of the adverb (*definitely*), associated with a positive 'preference' verb (*want*). This verb seems more committed to preference than a verb such as "*(to) like*", which might appear as an opinion verb. Commitment can be a clue for preference in the sense of decision making, and extend the simple vocabulary of opinions. Strong semantics have been criticised since it may lead to contradictory preferences when several preference statements are considered.

Ceteris paribus semantics slacken these requirements by adding a further constraint upon the variables that are not concerned in the preference. Thus α-outcomes are ordered above β-outcomes, only if they are completed by the same variable assignment for all other variables. We use here the most common and used one. Taking up the same example, a ceteris paribus preference for London over Paris would result in ordering London Universities above Paris Universities only when they have the same rank. The NL form could be "*at equivalent rank, I prefer London universities over Paris universities*". This can be seen as a particular case of a *conditional preference*. Terms such as '*equivalent*' in a dependent chunk (not qualifying a precise variable) or expressions such as '*all being the same*' are good clues for a ceteris paribus preference.

Although strong and ceteris paribus semantics are the most natural for expressing preferences, they do not leave much room for exceptions. This makes them unsuitable to reason about defeasible preferences. Optimistic and pessimistic semantics have been proposed in non-monotonic reasoning to deal with defeasible knowledge. By further relaxing requirements, these semantics allow for exceptions. An optimistic preference of London over Paris could result not only in ordering London universities over Paris ones, but also ordering non-Paris universities, say Berlin universities, above Paris universities. Using a pessimistic preference of London over Paris, we can have a university ordering where London universities are preferred not only to Paris universities, but also non-Paris ones. When associated with the object of preference, *commitment verbs* could suggest an optimistic preference (e.g. "*I would go for a London university any day*") and *negations*, a pessimistic one (e.g."*It will certainly not be a Paris university*").

Outline of the Chapter

To reach our objective of acquiring user preferences to be described as comparative preference statements, we consider the linguistic, psychological and behavioural aspects indicated above, and settle upon the aim of formalising a protocol for acquiring user preferences that can lead to their representation as comparative preference statements.

This chapter first describes a linguistic study conducted to (1) analyse the nature of NL-expressions that convey user preferences, (2) look for the evidence of comparative preference statements and accompanying semantics in their meaning and (3) develop a linguistic framework for identifying them from textual corpora. Next, using *preference elicitation* as an adapted paradigm for acquiring such preferences, it describes a protocol for preference elicitation, building the linguistic resources it requires.

The key questions these raise are the following: (1) Are preference linguistic patterns different from opinion expressions, when faced with preference semantics theories? (2) Does retrieving them require specific corpora, i.e. dialogue corpora since elicitation is a dynamic process, and if so, what are the linguistic clues denoting preference expressions? (3) Can natural language processing help in improving the elicitation process by increasing the accuracy of the interaction with the user? This study tries to shed light on these questions by:

1. assessing the specificity of preferences over opinions while relying on existing literature in preference elicitation, and providing requirements for an appropriate corpus for analysis (§2.1);
2. searching for preference textual clues within the corpus to build a lexical framework to identify preferences, matching the different semantics with their linguistic forms using a set of templates (at the sentence level) (§2.2);
3. evaluating the improvement through a set of requirements for an elicitation protocol tested in a crowd-sourcing experiment (§2.3 and §2.4).

2.1 Preferences, Opinions and their Respective Corpora

From a linguistic point of view, preferences and opinions are judgmental assertions and thus tend to look alike on the surface, but often differ when it comes to their operational aspects. Moreover, expressions in natural language can often mean the same thing but leave room for different interpretations (e.g. “I am not a big fan of” could be, but is not always, equivalent to “I dislike”).

A first step is therefore to analyse corpora containing such expressions and identify markers that distinguish preferences and opinions from each other.

2.1.1 Preferences vs Opinions

Preferences and opinions don't have the same purpose or effect. Opinions reflect a personal evaluation of an object and do not necessarily influence a decision-making process. Preferences, on the other hand, are expressed to simplify decision-making as they imply a rank-ordering of outcomes and help create a model of the user's likes and dislikes in the specific context of a choice problem.

Linguistic resources to identify opinions have been developed by analysing corpora of user reviews. Such corpora contain the attributes about which a user may (or may not) have a preference, but not the nature of the preference itself. They can thus be used to predict possible preferences, but not analyse them. As a consequence, they should be unsuitable to identify preference statements.

To verify our intuition, we have built our own resources to identify preferences from an appropriate corpus, and applied both resources on the same corpus to estimate the coverage of preference terminology by opinion terms. The comparative analysis is detailed in 2.4 on page 52.

2.1.2 Requirements for an Appropriate Corpus

Previous research in text mining has tackled the problem of identifying comparative sentences in text documents (Jindal and Liu, 2006). In our quest for corpora containing preference statements, we analysed their corpus of labelled comparative sentences to discover forms of NL preferences within these annotated comparative sentences.

There exist examples of user preferences within the corpus, but, as confirmed by the authors, these express a comparison indirectly through preferences. Taking up one of their examples, "I prefer Intel to Amd" has been found in the corpus, but is actually intended to express "Intel is better than Amd" in a comparative sentence. The former could be seen as an NL preference, but since it is not expressed within the context of a choice problem, it does not imply an ordering of all possible outcomes. If, instead, the same sentence was uttered in the context of buying a laptop, it would affect the ordering of laptops according to the user's preference of Intel over Amd. This makes the corpus unsuitable for a semantic analysis of the preferences expressed. We also speculate that a dialogue corpus leading up to a user making a choice could provide a better means of analysing NL preferences.

We found an existing corpus, the corpus PLUS (Pernel, 1991), that fulfils this last requirement. It is aimed at creating a human-computer written dialogue system, containing short and to the point conversation transcripts about yellow-pages directory enquiries, in a non-digital format. Prince and Pernel (1994) have analysed this corpus of 71 dialogues (with 7-49 turns) according to dialogue structure and demonstrated its importance in choice elicitation. We performed a manual analysis to detect preference expressions and found 206 NL expressions. This allowed us to conjecture a lexical base for preference expressions. Since it was a french corpus, we developed a prototype preference lexicon using french terms. Since we wanted to conduct experiments in English, we translated this lexicon into English to test its reliability.

Having pointed out the distinctions between preferences and opinions, and how their respective corpora differ, we now describe how we adapt existing intelligent text processing techniques to identify preferences within appropriate corpora.

2.2 Linguistic Framework for Identifying Preferences

Our model for identifying preferences from textual resources is based on existing methods in information extraction. We begin with a quick review of such methods to determine those which could best be adapted to extracting and representing preferences using conditional logic semantics.

2.2.1 Background: Intelligent Text Processing

Acquiring preference data from textual resources (e.g. web pages, blogs, fora, dialogue transcripts, etc.) could be performed by:

1. constituting a corpus of texts (speech, dialogue transcripts) or discussions (as could be found in a forum or a blog) which contain preferences;
2. attempting to identify regular forms that could correspond to preferences; and
3. extracting them in order to populate a dedicated preference database.

Everything we have seen so far regarding preferences leads us to conclude that their extraction from a textual corpus would require a complex procedure involving several NLP techniques. This would mean that what we seek here is not only text processing, but intelligent text processing–the keyword being 'intelligent'.

Essentially, the various technologies could be broken down into categories such as (1) Syntax and Parsing, (2) Semantics and Dialogue (3) Information Extraction.

Syntax and Parsing. In this category, the approach is focussed on the grammatical syntax and parts of speech (POS) in sentences.

Semantics and Dialogue. Here, the meaning of words and their correlation with other words are exploited, to extract dedicated terminology.

Information Extraction. Here, structured information is automatically extracted from unstructured texts. This often requires the combination of different NLP techniques to achieve its aim. Examples of such work are seen in the web-based information extraction systems.

Amongst these technologies, those that can be adapted for extracting preferences are those that favour extracting *linguistic patterns* as opposed to those that use *semantic parsing* in order to extract terminology.

Textual Clues There is a notion amongst linguists (computational and otherwise) called *textual clues* (Péry-Woodley, 1993) which allows objects to be identified on the basis of certain common traits (e.g. linguistic forms, set expressions, semantic key-words, etc.) they exhibit. They have been used in models that seek to identify cognitive processes such as argumentation (proposed by Moeschler (1989)), text summarising (by Charolles (1990)), account of scientific experiments (by Lucas (1993)) and explanations (by Prince (1996)).

There exist hybrid methods using textual clues as well, such as proposed in Prince (1999) of (1) identifying *textual clues* in the cognitive process of explanations and (2) selecting *relevant patterns* within these, in order to create a model for the analysis and production of the extracted explanations.

We believe that this method is structurally adaptable to our purpose. Textual clues in the expression of a preference could be identifiable by a similar process and manipulated using structures that correspond to preference representations in conditional logic. We therefore take a closer look at their proposed model.

An Approach for Analysing Explanations The contribution (Prince, 1999) describes a protocol for corpus analysis which constitutes a necessary preparatory phase for the actual implementation of an automated system which analyses and produces explanations. This protocol focusses on recognising and classifying linguistic, structural and semantic clues in explanatory texts, thereby detecting the interlocutor's intention behind them. These textual clues, when modelled in the form of text "templates" (or programmable structures), can subsequently be employed by the system so that the user is directly informed of the explanatory intent, facilitating their apprehension of the automatically generated explanations.

In the following paragraphs we trace the broad outlines of the method proposed in this article, concluding with an illustration of how we intend to adapt it for our objective.

Step 1 - Identifying Textual Clues Textual clues can be characterised by three main attributes: a lexical marker, a semantic keyword, and a syntactic structure related to the grammatical category of the latter. In Prince (1999), these are specifically determined in order to detect explanations in the corpus. The explanations themselves have been categorised according to their communicative intent, thus different textual clues are attributed to each of the distinct categories. Table 2.1 shows these different categories with the textual clues that are expected in the text while analysing the corpus.

Communicative Intent	Textual Clue Expected
Explanation by potentiality	Modal verbs (can, have, must, etc.)
Explanation by particularity	Assertion of the property describing the object in question
Implicitly explanatory	Lack or absence of textual clues
Explicitly explanatory	Presence of more than one textual clue

Table 2.1 – Communicative Intent and Corresponding Textual Clues

Once these clues have been identified in the text, the next step is to try and formalise them.

Step 2 - Deriving Programmable Structures Each type of textual clue is distinguished by its form and named according to its function, the latter being specific to the cognitive process of explanation. The term used for these structures is *"explanation variables"*. Their form is defined as a *"canonical form"*. Table 2.2 shows these variables and some of their corresponding canonical forms [2].

Explanation Variables	Template (Canonical Form)
Justificatory	[A] <parce que> {CAUSE_DE A}
Argumentative	<annonce-argumentative> [EXPLICATION] {ARGUMENT}
Illustrative	<annonce-illustrative> [EXPLICATION] {EXEMPLE}
By definition	[A] <est> {PROPRIETES_DE A}
By inference	<SI> [A (est observé)] <alors> [PLAN D'ACTIONS]
	<SI> [A (est vrai)] <alors> [CONSEQUENCES]
Plausible	[A] peut être [SYMPTOMES DE A]

Table 2.2 – Categorised Textual Clues and Corresponding Templates

2. The canonical forms are given here in French as the work was done on a French corpus. We feel it is appropriate to keep them in their original language as translating them would diminish their authenticity.

Interestingly, the corpus analysis using explanation variables gives rise to the recognition of well-known categories of explanations such as *justificatory, argumentative and illustrative*. Given the minutely specific approach of textual clues, this could very well have been lost. The fact that it has *not* gives us hope for our own application of textual clues for the identification of preference expressions and reinforces our decision to pick this form of NLP.

Finally, the study is rounded up with an explicit correspondence established between the communicative intent behind an explanation and the canonical forms identified as a result of corpus analysis. This defines the heuristic techniques that can eventually be implemented in an automated system generating explanations. Table 2.3 lists some of these techniques [3].

Communicative Intent	Template (Canonical Form)
Explanation by potentiality	[A] peut être [SYMPTOMES DE A]
	[A] <parce que> {CAUSE_DE A}
Explanation by particularity	[A] <est> {PROPRIETES_DE A}
	[A] <parce que> {CAUSE_DE A}
Implicitly explanatory	- (absence of textual clues)
	...[A (vrai/observé)] ...[CONSEQUENCES/ACTIONS]
	{CAUSE_DE A}
	[A] [SYMPTOMES DE A]
Explicitly explanatory	<SI> [A (est observé)] <alors> [PLAN D'ACTIONS]
	<SI> [A (est vrai)] <alors> [CONSEQUENCES]
	<annonce-argumentative> [EXPLICATION] {ARGUMENT}
	<annonce-illustrative> [EXPLICATION] {EXEMPLE}

Table 2.3 – Communicative Intent and Corresponding Templates

Summing Up The approach described in this section treats explanations precisely in the way we wish to treat preferences. In Prince (1999), a thorough analysis of explanations leads to their semantic classification, each class of which is eventually converted into a programmable structure. We seek to achieve this very same goal with preferences.

3. The words in italics are in French as the work was done on a French corpus. We feel it is appropriate to keep them in their original language as translating them would diminish their authenticity

	Explanations	Preferences
Textual clues	*verbes modaux*, etc.	Verbs denoting preference (prefer, favour, etc.)
	car, parce que, puisque, etc.	Comparative adverbs (more than, less than, etc.)
Programmable structures	justificatory, argumentative, etc.	strong preference, conditional preference, comparative preference etc.

Table 2.4 – Theoretical Adaptation from Explanations to Preferences

The programmable structures found correspond to categories in explanation which are well-known in the literature (justificatory, argumentative and illustrative). Similarly, we would like to find the correspondence between the preference classification that we find through corpus analysis and the pre-existing one in conditional logic. Table (2.4) shows how we intend to adapt this approach to our own.

2.2.2 Our Approach: An Outline

Our linguistic analysis of preference expressions is based on the approach described above for explanations. We analyse the corpus PLUS to discover textual clues for identifying preferences and in the process, develop our preference lexicon.

Our approach relies on the two following principles: (1) the inherent structure of dialogue is essential to the proper elicitation of preferences and (2) textual clues which can be identified within the transcripts are instrumental to a further analysis and classification of the preferences extracted. Keeping this in mind, the model we define follows a three-phase procedure:

1. *Analysing and annotating* the corpus to determine how and where preferences are expressed within a dialogue framework;
2. *Identifying keywords* within these annotated preferences to ascertain their semantic properties;
3. *Deriving programmable structures, or preference templates,* out of identified expressions so that they may be classified by type and eventually represented formally using conditional logic.

2.2.3 Step 1 - Corpus Annotation

When analysing the dialogue corpus, we were rapidly made aware of the strong role played by the dialogue structure in itself. We found that when users are confronted with a question, their answers are short and to the point. This is distinctly different from when they write (as found in the text corpora), for in this latter case they have time for reflection and perfect formulation. Moreover, the very structure of the dialogue gives a greater control to the compère in orientating the conversation towards the elicitation of preferences.

Thus, our annotation scheme is designed to make full use of this structural advantage. Borrowing from the dialogue processing method proposed in Moeschler (1989) for argumentative inference, our annotation scheme operates in three phases: (1) preliminary context analysis, (2) structural breakdown of conversations, and (3) final annotation.

Preliminary Context Analysis. Our first step is to study how the context in which these dialogues were conducted contributes to their meaning and organisation. The fact that all the dialogues collected within this corpus share the same aim of interrogating a database of yellow pages in the Parisian district allows us to make the following underlying assumptions about the text we annotate:

— All dialogues begin with an enquiry and end when it is answered (successfully or not).
— They follow a scheme of question-answer pairs, often nested.
— The compère counter-questions the subject to elicit their preferences until their initial enquiry is sufficiently refined; only then does he consult the database.

The first two assumptions help perceive a common organisational pattern in the conversations, leading to a more comprehensive annotation of the corpus. The last assumption is particularly helpful to our cause as it provides a necessary heuristic for locating preferences.

Structural Breakdown. The next phase in the annotation scheme is to break each conversation down according to question-answer pairs, revealing nested segments when they exist. Thus, instead of treating the text in a linear fashion, we simplify and improve our annotation by taking contextual information into consideration. We now have comprehensive snippets of dialogue, each one detailing a particular aspect of the conversation.

Final Annotation. In this last phase of the annotation scheme, we mark these snippets of dialogue according to their function. This will enable us to locate preferences within them. Every question-answer pair is annotated with the following markers:

Enquiry-Solution Markers A typical conversation contains two types of question-answer pairs. The first leads to a satisfactory refinement of the initial enquiry and the second leads to a satisfactory refinement of its resolution. Thus, within these pairs, we annotate all those statements made by the subject detailing or modifying an enquiry as "`Enquiry`" and all those statements made by the compère detailing or modifying a solution as "`Solution`". Here is an example from the corpus:

Example 1. [4] S: Je pars en voyage, je cherche vous prendre une assurance adaptée aux risques des pays tropicaux. `<Enquiry>`

[...]

C: Vous avez: Mutuelles d'assurances; Tel: 01 42 79 12 12; 29 Bd Edgar Quinet 14e. `<Solution>`

Attribute-Preference Markers In the conversation, when the compère finds that the enquiry is not precise enough, she/he/it asks the subject further questions in an attempt to elicit their preferences on the matter. These correspond to nested question-answer pairs. The nested question typically is about a certain attribute concerning the object of enquiry; the answer to which generally is the expression of a preference. We annotate the former as "`Attribute 1`" `(category)`, "`Attribute 2`" `(category), etc.` and the latter as "`Preference 1`" `(category)`, "`Preference 2`" `(category), etc.`. An example from the corpus makes this easier to see:

Example 2. S: Je voudrais aller manger à Paris. `<Enquiry>`

C: Dans quel quartier? `<Attribute 1 (location)>`

S: Dans l'ouest. `<Preference 1 (location)>`

C: Avez-vous une préférence au niveau de la cuisine? `<Attribute 2 (cuisine served)>`

S: Je voudrais quelque chose qui sorte de l'ordinaire. `<Preference 2 (cuisine served)>`

Preference expressions can already be spotted after this preliminary annotation. Table 2.5 shows some statistics concerning the annotation process.

4. The text in the example is expressly left uncorrected as the corpus itself often contains errors. These had been left as they are for evaluation purposes when the corpus was constructed.

	Total in corpus	Average per dialogue
Dialogues annotated in corpus	71	
Enquiry markers	76	1.07
Solution markers	162	2.13
Attribute markers	245	3.45
Preference markers	206	2.90

Table 2.5 – Annotation Statistics

In the next step of our proposed approach, we work on the preferences thus annotated and with the help of textual clues, identify their nature.

2.2.4 Step 2 - Identifying Textual Clues

The first task at hand to identify the nature of preferences, is to construct a specialised dictionary that covers all the linguistic elements that can be associated with expressing preferences in natural language. In other words, this dictionary contains an inventory of the textual clues that are to be expected within a preference expression in the dialogue corpus.

Two kinds of textual clues were found in the corpus: (1) verbs indicating the presence of preferences and (2) descriptors (adverbs, adjectives and adverbial locutions) elaborating their meaning.

verbs	descriptors
prefer, need, require, select, favour, distinguish, opt, choose, adopt,	more, very, strong, too, best, many
elect, sort, designate, take, name, promote, support, encourage, privilege,	well, better, most ,also, moderately, relatively
want, like, lean towards, help, provoke, isolate, reward, levy, seize, see,	barely, as, almost, rather, hardly,
differentiate, single out, characterise, discriminate, recognise, discern, note,	less, not at all, least, worse, lesser, not,
dissociate, notice, separate, set, determine, decide, adjudicate, vote, pick,	necessary, particular,
go (for), settle on, wish.	above, under, around, below.

Table 2.6 – Lexical Base for the preference lexicon. Descriptors are adjectives, adverbs and adverbial locutions.

We thus made a prototype preference lexicon containing 45 verbs in their base forms and 29 descriptors (see Table 2.6 on the preceding page). These keywords were further categorised semantically using 7 attributes which can be seen as 3 sets of mutually exclusive attributes (See Table 2.7). The first set in this table pertains to the *comparative* nature of preference expressions, i.e. when a preference is expressed by explicitly comparing objects. The second set refers to the *polarity* of preference expressions, i.e. if they are affirmations, or negations. The third set *qualifies the intensity* attributed to the preference expression. Thus, a given keyword can have a maximum of 3 attributes at the same time.

Attribute	Definition	Example
difference gauge	Evaluates the difference between objects.	I like coffee more than tea.
similarity gauge	Evaluates the similarity between objects.	I like coffee as much as tea.
modifier	Modifies the polarity of the expression (negations).	I'm interested. *becomes* I'm not interested.
enhancer	Enhances the polarity of the expression (has no polarity of its own).	"If it is extremely fast." *is positive, while* "If it is extremely slow." *is negative*.
qualifier-high	Lends a pronounced degree of intensity to the preference.	I am very much inclined.
qualifier-medium	Lends an average degree of intensity to the preference.	I am fairly inclined.
qualifier-low	Reduces the degree of intensity of the preference.	I am less inclined.

Table 2.7 – Defining the Attributes

2.2.5 Step 3 - Identifying Preference Templates

This last and final step in our analysis is aimed at exploiting the information gathered during the two preceding phases, to organise the extracted preferences according to their semantic properties. This is achieved by sorting our descriptors (semantic keywords) with respect to their attributes, using Formal Concept Analysis (FCA) techniques proved to be efficient according to Priss and Old (2004).

Before getting into the details of how this contributes to the sorting of the descriptors, we shall look at some preliminary definitions concerning FCA.

FCA - Preliminary Notions Below are some simplified definitions for aspects in FCA which we shall employ for our study.

Definition 7 (Formal Context)**.** A *formal context* consists of a set of objects O, a set of unary attributes A, and an indication of which objects have which attributes. It may be described as a table, with the objects corresponding to the rows of the table, the attributes corresponding to the columns of the table, and a Boolean value in cell (x, y) whenever object x has value y.

In our case, O is the set of descriptors (semantic keywords), A is the set of attributes defined in Table 2.7 and the context is a table with the Boolean value represented as a cross.

Definition 8 (Formal Concept)**.** A *formal concept* for a context is defined to be a pair (O_i, A_i) such that

1. $O_i \subseteq O$
2. $A_i \subseteq A$
3. every object in O_i has every attribute in A_i
4. for every object in O that is not in O_i, there is an attribute in A_i that the object does not have
5. for every attribute in A that is not in A_i, there is an object in O_i that does not have that attribute

O_i is called the extent of the concept, A_i the intent.

Definition 9 (Concept Lattice)**.** The concepts (O_i, A_i) defined above can be partially ordered by inclusion: if (O_i, A_i) and (O_j, A_j) are concepts, we define a partial order $\leqslant$ by saying that $(O_i, A_i) \leqslant (O_j, A_j)$ whenever $O_i \subseteq O_j$. Equivalently, $(O_i, A_i) \leqslant (O_j, A_j)$ whenever $A_j \subseteq A_i$. Thus ordered, they are represented in the form of a lattice, called the *concept lattice*.

Sorting Descriptors using FCA We created a formal context defining the relation between the keywords and their attributes. Generating a concept lattice [5] out of this context produced a terminological base revealing a classification based on the attributes. This sorting method allows new words to be appended to the list and automatically classified by regenerating a lattice.

	difference	similarity	modifier	enhancer	qualifier-...	qualifier-...	qualifier-l...
very				X	X		
extremely				X	X		
exceedingly				X	X		
much		X			X		
greatly				X	X		
a lot				X	X		
slightly			X			X	
somewhat			X			X	
less	X		X				X
not			X				
more	X				X		
worse							
poorly			X				X

Figure 2.1 – Descriptors–Portion of Formal Context

5. using the open-source FCA software `ConExp (http://conexp.sourceforge.net)`.

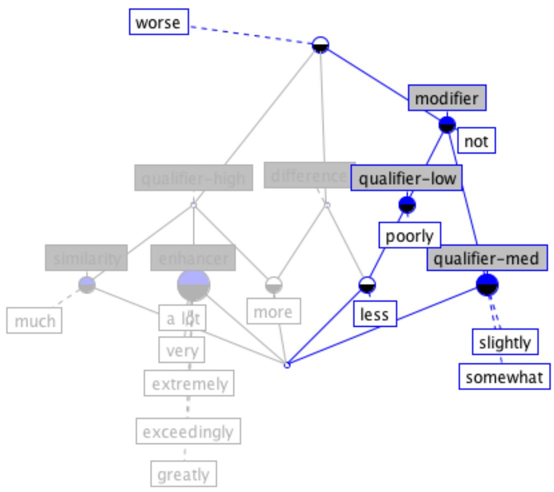


Figure 2.2 – Descriptors – Portion of Concept Lattice

Figures 2.1 and 2.2 present portions of the context and its lattice. The lattice shows how keywords are classified according to a hierarchy: the highlighted portion shows how the keywords are all 'modifiers' but they sub-classify according to the attributes 'qualifier-med' and 'qualifier-low'. The complete lattice for our prototype lexicon contained 43 concepts.

A significant benefit of using this sorting method is that it is language independent. Simply replacing the keywords with the corresponding word in another language preserves the lattice structure and thereby the semantic classification. We could test this by analysing the lexicon coverage for the french corpus PLUS and our own english corpus from the crowd-sourcing experiment. Of the 29 keywords in our lattice, 24 were found at least once in the french corpus and 17 in the english one.

Identifying Preference Templates As we saw in §2.2.1, programmable structures are defined using a canonical form. We attempt to do the same here, using the textual clues described above. Our approach consists of forming patterns or templates using the keywords from our preference lexicon to construct a preference expression.

Preference Template	Description	Preference Type	Example
Absence of verbs	Direct answer for the attribute described in the question. Short and to the point.	Direct	Q: "Is ultra-portability a necessary component for your laptop?" A: "[Not very]" or A: "Yes"
[subject] <verb> [object of preference]	Presence of preference verb and instantiation of attribute.	Basic	"[The laptop] <needs to have> [good speakers]"
[subject] <verb> [object of preference 1] <adverb> [object of preference 2]	Presence of synonym and one value of attribute compared with another.	Relative	"[I] <would prefer> [Windows 8] <to> [MacOS]"
If [condition] + {basic template}	Presence of keyword 'if' with the basic preference template.	Conditional	"{It's not as important} if [I own an external hard drive]."

Table 2.8 – Preference Templates and their Categories

Strong	Ceteris Paribus	Optimistic/ Pessimistic
- Direct/Basic preference template with presence of 'enhancer' and 'qualifier-high' keywords (LT)	- Presence of terms such as 'all else equal', 'when it comes down to', 'everything else being the same', etc. (LT)	- Conditional preference template with contradictory preferences. (I) - Relative preference template with descriptors such as 'but', 'however', etc. (LT) - Presence of 'enhancer' and 'qualifier-high' keywords (Pess.) and 'modifier' keywords (Opt.). (LT)

Table 2.9 – Linguistic Terms (LT) and Indicators (I) associated with Preference Semantics

The NL expressions are thus distinguished by type, which eventually leads to their representation using comparative preference statements. Table 2.8 presents the templates and corresponding preference categories with an example from the corpus collected from the crowd-sourcing experiment serving as proof of concept (§2.4). Table 2.9 shows how the templates can be associated with different preference semantics.

It is interesting to note that the *bipolar correspondence between optimistic and pessimistic semantics is also reflected in this table*: 'enhancer' and 'qualifier high' keywords reflect pessimistic semantics (i.e. positive preferences), and 'modifier' keywords reflect optimistic semantics (i.e. negative preferences)

This completes our linguistic analysis of a dialogue corpus of preference expressions. Equipped now with a dedicated preference lexicon and preference templates that lead to identify comparative preference statements with associated preference semantics, we can address the primary aim of our chapter: the bottleneck in preference acquisition methods, especially regarding qualitative preference representation languages.

In the next part of this chapter, we develop a protocol for preference elicitation and evaluate it using a crowd-sourcing experiment as proof of concept. This allows users to express their preferences in natural language, and the system to represent them as comparative preference statements.

2.3 Protocol for Preference Elicitation

Our approach to identifying NL preferences leads to the development of a preference lexicon and accompanying preference templates to distinguish different forms of preferences, each corresponding to those represented as comparative preference statements. We now present some observations about this study, and motivations for how this could lead to developing a protocol for preference elicitation.

Motivations. Our linguistic study revealed the existence of NL preference expressions (implicit and explicit) that could be represented as comparative preference statements with associated semantics. It provides a very concrete way to analyse the process of decision-making using structured preferences. Its most important contribution is its interrogative structure, as this gives the compère a degree of control which is necessary for an accurate elicitation of preferences.

By restricting the nature of questions to those that would lead to a proper elicitation of relevant user preferences, it should be possible to represent them formally using comparative preference statements, so that the best choices obtained from the induced preference relation are made available to the user in their decision-making process. In particular, careful attention must be paid to the elicitation of hidden preferences.

This is because linguistic forms for semantics other than ‘strong’ were not found in this corpus, although several examples showed their possible existence, had they been elicited by the compère. We describe one such example in detail, as it shows an implicit existence of preferences of varying importance, and the relevance of eliciting preferences other than strong preferences. [6]

Example 3.

S: Je voudrai aller manger à Paris.

C: Dans quel quartier ?

6. Annotation markers have been omitted and those parts of the text which reflect preferences have been accentuated for better readability.

S: Dans l'ouest

C: Avez-vous une préférence au niveau de la cuisine ?

S: Oui

C: Laquelle ?

S: Je voudrais quelque chose qui sorte de l'ordinaire

C: Vous voulez une cuisine étrangère ?

S: Oui, exotique mais pas ordinaire

C: Veuillez patientez quelques instants

C: Pouvez-vous préciser le pays ?

S: Quelque chose de pas trop épicé.

C:

1. A la banane ivoirienne
   01 43 70 49 90
   10 r Forge Royale, 11e
2. Antioche rest. greco-turc
   01 42 52 03 75
   12 r Doudeauville, 18e
3. Au palais de l'Himalaya
   01 43 70 49 90
   2 r Briquet, 18e
4. Au vieux Budapest
   01 46 33 09 51
   40 r Descartes, 5e

S: Oui, le restaurant hongrois

C: Si vous n'avez plus de demande, au revoir

S: Ce restaurant n'est pas dans l'ouest de Paris

C: Excusez-moi, je n'ai pas une très bonne memoire

C: C'est le seul restaurant hongrois de Paris.

S: Bon, alors un resto russe

C: Veuillez patientez, merci

C: Le madrigal, 01 40 43 99 96, 1-2 Sq Auguste Chabrières, 15e

S: Ok merci

C: Au revoir

Formal Representation:

$$V = \{\mathit{quartier}, \mathit{cuisine}, \mathit{lieu}\};$$
$$\mathit{Dom}(\mathit{quartier}) = \{\mathit{ouest}, \neg\mathit{ouest}\};$$
$$\mathit{Dom}(\mathit{cuisine}) = \{\mathit{exotiqueOrientale},\ \mathit{exotique}\neg\mathit{Orientale}\};$$
$$\mathit{Dom}(\mathit{lieu}) = \{\text{épicé}, \neg\text{épicé}\};$$

with the following preferences:

$$\mathit{ouest} > \neg\mathit{ouest};$$

$$\mathit{exotique}\neg\mathit{Orientale} > \mathit{exotiqueOrientale};$$
$$\mathit{exotique}\neg\mathit{Orientale} \wedge \neg\text{épicé} > \mathit{exotique}\neg\mathit{Orientale} \wedge \text{épicé}.$$

Based on these preferences extracted from the example, we can deduce that the compère had left the user's preference of 'ouest' out of account when proposing the first list of restaurants (amongst which the user picked the hungarian "Au vieux budapest") because they focussed on finding restaurants only based on the cuisine they served. As the user then revealed a preference for 'russe' which is in the west, it shows that they had a stronger preference for 'ouest' than for '$\mathit{exotique}\neg\mathit{Orientale} \wedge \neg\text{épicé}$'.

Now, if the compère had further questioned the user about the importance of the different preferences elicited, the acquired preferences could have been treated using ceteris paribus, optimistic or pessimistic semantics and consequently this problem would have been averted.

What we mean by this is that using preference semantics while stating a problem offers a necessary advantage for its resolution. We demonstrate by considering the preferences stated in this problem using four semantics from conditional logic: strong, ceteris paribus, optimistic and pessimistic. We then manually compute the preference relation induced by the set of preference statements, and discuss how our results prove that the latter three semantics offer a solution while strong semantics fail to do so.

Formally, we have: $\Omega = \{$

$\omega_1 = \mathit{ouest} - \mathit{exotiqueOrientale} - \text{épicé}$

$\omega_2 = \mathit{ouest} - \mathit{exotiqueOrientale} - \neg\text{épicé}$

$\omega_3 = \mathit{ouest} - \mathit{exotique}\neg\mathit{Orientale} - \text{épicé}$

$\omega_4 = \mathit{ouest} - \mathit{exotique}\neg\mathit{Orientale} - \neg\text{épicé}$ (this corresponds to 'russe')

$\omega_5 = \neg\mathit{ouest} - \mathit{exotiqueOrientale} - \text{épicé}$

$\omega_6 = \neg\mathit{ouest} - \mathit{exotiqueOrientale} - \neg\text{épicé}$

$\omega_7 = \neg\mathrm{ouest} - \mathrm{exotique}\neg\mathrm{Orientale} - \mathrm{épicé}$

$\omega_8 = \neg\mathrm{ouest} - \mathrm{exotique}\neg\mathrm{Orientale} - \neg\mathrm{épicé}$ (this corresponds to 'hongrois')

}

Based on the user's preferences extracted from the corpus, we have
$\{\omega_1, ..., \omega_4\} > \{\omega_5, ...\omega_8\}$;
$\{\omega_3, \omega_4\ \omega_7, \omega_8\} > \{\omega_1, \omega_2\ \omega_5, \omega_6\}$;
$\{\omega_4, \omega_8\} > \{\omega_3, \omega_7\}$.

Treating these preferences respectively using strong, ceteris paribus, optimistic and pessimistic semantics, we manually compute the preference relation $\succeq$ following algorithms proposed for this purpose in the literature (Kaci, 2011) (which shall be further discussed in the following chapter).

Given $\succeq = (E_1, ..., E_n)$ such that $E_i > E_{i+1}$, we have

Strong (method 1): $E_1 = \omega_4, E_2 = \emptyset \implies$ Inconsistent preference statements, but can be exploited to reveal ω_4 as most preferred.

Strong (method 2): $E_1 = \emptyset, E_2 = \{\omega_5, \omega_6\} \implies$ Inconsistent preference statements with no preferred solution.

C.P. (method 1): $E_1 = \{\omega_4\}, E_2 = \{\omega_3, \omega_8\}, E_3 = \{\omega_1, \omega_2, \omega_7\}, E_4 = \{\omega_5, \omega_6\}$.

C.P. (method 2): $E_1 = \{\omega_4\}, E_2 = \{\omega_3, \omega_8\}, E_3 = \{\omega_1, \omega_2, \omega_7\}, E_4 = \{\omega_5, \omega_6\}$.

Optimistic: $E_1 = \{\omega_4\}, E_2 = \{\omega_1, \omega_2, \omega_3, \omega_5, \omega_6, \omega_7\}$.

Pessimistic: $E_1 = \{\omega_4\}, E_2 = \{\omega_1, \omega_2, \omega_3, \omega_7\}, E_4 = \{\omega_5, \omega_6\}$.

This shows that the preference relations computed when considering preference semantics other than strong semantics result in each case with ω_4 (russe) as being the most preferred alternative, the solution which the user themselves chose in the dialogue. Thus, had the implicit preference been elicited by the compère from the start, they would not have proposed 'hongrois' as a viable restaurant option.

We conclude that in order to elicit the various preference semantics, the compère must ask further questions to reveal some of the subject's hidden preferences, such as the inherent priorities the user has about their different preferences, as revealed by this example.

Our Protocol for Preference Elicitation. We now propose a protocol for NL preference elicitation, retaining the qualitative aspects of this study, and demonstrate its feasibility through a crowd-sourcing experiment (2.4 on the following page).

To ensure preference elicitation for customised decision support, the protocol must be adapted to the specific decision-making scenario. This is done by: (1) forming a database of outcomes to choose from, (2) fixing a number of attributes which correspond to the different aspects of the outcomes and (3) for each of these attributes, designing a specific question to elicit the user's preferences.

The content of these questions determines the linguistic elements we look for in the answers (e.g. quantitative attributes such as 'size' would have quantitative descriptors in the answer). When the answer contains implicit or ambiguous preferences, further questions are asked to reveal them explicitly.

When expressions concern defeasible preferences in particular (e.g. contain words such as "but" or "however"), optimistic or pessimistic semantics can be used to resolve them. For example, "I prefer Windows to MacOS, but if the laptop has a small screen, then I prefer MacOS to Windows." suggests a default preference for Windows over MacOS which is reversed. Here, further questions need to be asked to ascertain which of optimistic or pessimistic semantics suits the user. Accordingly, if laptops with Chrome OS (resp. with Chrome OS and a small screen) are an acceptable choice for the user w.r.t. "prefer Windows to MacOS" (resp. "prefer MacOS to Windows if the laptop has a small screen"), then optimistic semantics is applied. Otherwise, pessimistic semantics is used.

Once all the attributes are instantiated, the elicitation is complete.

2.4 Evaluation by Proof of Concept

To demonstrate the feasibility and subsequent utility of the preference elicitation protocol developed in the previous section, we adapted it to choosing a laptop, and implemented it using PyBossa[7], an open-source crowd-sourcing development framework, as a proof of concept.

We were able to collect sufficient data (56 contributors) for a valid evaluation of the existence of preference semantics in NL expressions. For the purpose of this experiment, we manually designed a set of 9 questions (Table 2.10) which would cover the range of attributes necessary for choosing a laptop. The attributes were selected according to the technical specifications of laptops. For each attribute, we formulated a simple question which could be accessible to a layman, and would have three possible answer types: (1) Numerical values (2) Descriptive (3) Yes-No. This resulted in a corpus of 56 dialogues with 18 turns/dialogue.

We also analysed the corpus to assess the differences between preference and opinion terminologies. We note that this analysis was performed on a corpus containing the question followed by the answer. This was to take into account all the instances when the user provided a single word ('Yes-No') or a numerical

7. http://dev.pybossa.com

Question	Attribute	Answer Type
Let us suppose you wish to buy a laptop. What would your budget be?	Price	Numerical
Do you have a preference for a particular operating system (e.g. Windows 8, Mac OS X, Chrome OS)?	OS	Yes-No/Descriptive
What screen size are you looking for?	Screen Size	Numerical/Descriptive
Would you need to use your laptop for long hours without charging it?	Battery Life	Yes-No
Is ultra-portability a necessary component for your laptop?	Weight	Yes-No
How much data storage space do you need for documents, music, videos, etc.?	Hard Disk size	
Would you require a very powerful laptop (for applications like photoshop, sound editing, etc.)?	Processor Speed	Yes-No
What about intensive graphic use (e.g. 3D gaming, Video editing, etc.)?	Graphics Card	Yes-No/Descriptive
Is there something more you would like to specify?	Miscellaneous	Descriptive

Table 2.10 – Experiment: Questions and Corresponding Answer Types

answer, because in such instances, the preference terminology appears in the question (e.g. "Q: Is ultra-portability a <necessary> component for you? A: Yes").

As 'opinion terminology', we used (1) an English opinion lexicon (we call this O1) created in Hu and Liu (2004) containing 2006 entries, and (2) the comparative keywords (we call this O2) in Jindal and Liu (2006) containing 78 entries. Since these two contained many unknown and inflected words, we used TreeTagger Schmid (1994) to eliminate all unknown and inflected words from the two lists. This resulted in a final list of 572 verbs and 51 comparative keywords.

Comparing this with our preference lexicon, we found 16 common terms: *take, want, like, isolate, provoke, see, set, go* from O1 and *prefer, favour, choose, elect, least, less, more, most* from O2. There are comparatively very few common terms between O1 and the preference lexicon because the former contains a large number of sentiment verbs which rarely occur in preference expressions. As we can see, preferences focus largely on choices and therefore use choice verbs and comparatives.

Our next task was to use TreeTagger on our entire corpus to determine how these terms were reflected in preference expressions, in particular within the preference templates described in Table 2.9.

Results. The annotated data (using TreeTagger) was analysed using GATE [8]Cunningham et al. (2002) for a two-fold purpose: (1) identification of preference templates using a semi-automatic method and (2) comparative analysis of preference/opinion terms found.

For both purposes, the first task was to use a gazetteer to match words from the opinion and preference lexicons with tokens in the annotated corpus (this is the automatic part for the former). For the template identification, every token that was matched with the lexicons was analysed within its context to discover the presence, or not, of particular preference templates. For the comparative analysis, we compared the number of matches for terms from both lexicons (preference and opinion) in terms of their POS tags.

Preference Template Identification. We identified a total of 535 [9] preference templates, which showed the existence of the preference semantics. Table 2.11 summarises our results.

Template	(%) in Corpus	Associated Semantics
Direct	33.1	Strong / Ceteris Paribus
Direct (numerical)	33.6	Strict / Non-strict (any possible semantics)
Basic	18.1	Strong / Ceteris Paribus
Relative	8.9	Optimistic / Pessimistic
Conditional	5.2	Optimistic / Pessimistic

Table 2.11 – Templates Found and Associated Preference Semantics

In cases when many semantics could be associated with the same preference, further questions are required to determine the exact semantics. This was not performed during the current experiment. However, our current results are already an improvement from the results obtained with the previously annotated dialogue corpus: with 22% fewer dialogues, we obtained 2.5 times more preferences (206 vs 535) proving that NLP helped in improving the elicitation process by increasing the accuracy of the interaction with the user.

Since this experiment serves as a proof of concept, we have focussed on finding the existence of preference semantics in NL expressions. We have not deepened our analysis to evaluate the quality of our template identification process in this preliminary study.

8. An open-source text engineering environment (`http://gate.ac.uk`)
9. This number is higher than the number of answers (56 x 9 = 504) because in some cases the user provided several preferences. These are typically cases where further questions are required to refine the preferences.

The following are examples that illustrate the presence of textual clues and corresponding preference semantics found in the corpus:

1. *"Yes, graphics card needed* very much." The adverb 'very' combined with the determiner 'much' lends a pronounced degree of intensity to the verb 'need' associated with the preference expressed, making it a strong preference.
2. *"I hate windows 8 and I'm* not too *interested in Mac OS."* The combination of adverbs 'not' and 'too' modifies the polarity of the adjective 'interested' and the preference is rendered negative. Without the adverbs, it would have been positive, i.e. "I'm interested in Mac OS".
3. *"15 inch (Being the smallest) to 17 inch (*Bigger the better*)."* The adjectives 'bigger' and 'better' express what the numbers 15 and 17 alone do not: the user does not merely want a screen in the 15"-17" range, they want the biggest possible screen. This added preference could have an impact on the final outcome if there were two laptops to choose from, similar in every respect except that one has a 15" while the other has a 17" screen. This could be seen as a ceteris paribus preference.

Comparative Analysis: Preferences vs. Opinions. The corpus contained a very small variety of preference/opinion verbs. Moreover, most of the occurrences were verbs which were used in the questions, indicating an influence over the user's choice of words. Table 2.12 shows the figures of our comparative analysis.

Verb	# Occ.	Lexicon	Descriptor	# Occurrences	Lexicon
depend	16	P	more	66	P + O
need	110	P	very	60	P
get	2	-	too	10	P
go	1	P	not	66	P + O
like	59	P + O	necessary	56	P
prefer	57	P	particular	56	P
require	58	P	around	22	P
take	4	P	with	61	O
wish	54	P	only	8	O

Table 2.12 – Occurrence of Terms in Preference (P) and Opinion (O) Lexicons in Corpus

Comparing opinion terminology with preference terminology, we can say that they differ in terms of verbs, but are similar when it comes to adjectives and adverbs. This is in keeping with the notion that opinions differ from preferences w.r.t the user's actions but they are similar w.r.t the user's descriptions. Using O2, the list of comparative keywords, proved to be helpful for our task, as there were terms from this list which did not belong to our prototype preference lexicon, which were found within preference templates in the corpus.

This experiment provided a concrete means of testing our elicitation protocol. Compared to the dialogues examined during the preliminary corpus analysis, the elicited user preferences contained a larger number of preferences per dialogue and included a wider variety of preference types. This reflects the advantages we sought with the refinements added to our initial protocol and confirms the viability of our protocol. Thus, the data collected during the experiment constitutes a corpus of pure preference expressions in natural language, and can thus be analysed in future work to further enrich our preference lexicon and improve our preference templates.

Conclusion

There is a real bottleneck in preference handling in AI research w.r.t preference elicitation as it does not cater to the wide range of preference representation languages available, especially as regards languages which are based on comparative preferences and preference semantics. In response to this, as a first step in creating a decision-support tool using an AI based on such languages, we developed a preference lexicon and defined 'preference templates' to form a bridge between NL expressions and AI preferences. We then defined a protocol for preference elicitation which guides the user to express their preferences in NL and translates them into comparative preference statements. To complete the study, we implemented the protocol in a crowd-sourcing experiment which served as a proof of concept.

The results confirmed that the very nature of turn-by-turn dialogue provides an effectual structure for preference elicitation, something which prose (such as found in a textual corpus) does not fulfil with equal success. The latter contains numerous expressions of opinions, but very few preferences. It is the interactive nature of dialogue which reveals expressions of preferences. Our preference lexicon coupled with our preference templates, the two components of our linguistic framework for identifying preferences, have served to distinguish preference semantics in the NL-expressions elicited.

As a consequence of our crowd-sourcing experiment, we now have a corpus which contains authentic user preferences in natural language corresponding to comparative preference statements and their associated semantics. This provides a concrete link between natural language expressions and research in preferences in artificial intelligence.

This chapter provided a linguistic analysis of user expressions to confirm the relevance of using comparative preference statements and their associated semantics in a personalised decision support system. The next and crucial step leading towards the design of such a system is an in-depth analysis of comparative preference statements and their associated reasoning mechanisms. This will be the subject of our next chapter.

3

Comparative Preference Statements: A Closer Look

Introduction

We have seen where comparative preference statements lie in the landscape of research on preferences in AI and how they have not received extensive attention in existing decision support systems (Ch.1). We addressed part of the problem by proposing a protocol for preference elicitation to alleviate a bottleneck in preference acquisition methods (Ch.2). The second part of the problem lies in the efficient integration of its reasoning mechanisms into decision support systems. We address this issue in the present chapter.

To successfully integrate the advantages of using comparative preference statements into decision support systems, we must first have a thorough understanding of all the technical details of handling them. This is essential, whether or not these details eventually matter in the practical implementation of them.

We shall conduct our investigation of comparative preference statements by first going back to the grassroots of preference modelling (established in Ch.1) and building up our theory from there to formalising and reasoning with comparative preference statements, explaining every detail this involves, especially those that were previously omitted to maintain a global view on the theory of preferences in AI. This involves a discussion of the different semantics defined for comparative preference statements and the pitfalls and advantages of using each one of them (§3.1).

Next, we discuss the task of computing preference relations induced by sets of comparative preference statements and one or several semantics. We explain the existing reasoning mechanisms associated with these statements, and present algorithms developed in previous research for this purpose. This involves using a non-monotonic logic to reason with these statements. As we shall see, this is particularly tricky when given a set of statements which contains the use of several different semantics (§3.2).

We then conclude our investigation by looking into the behavioural aspects of the preference semantics and make a comparative analysis using postulates studied in preference logics and non-monotonic reasoning. We provide an extended version of the study presented in Kaci (2012a). Our selection of postulates is motivated by properties that could optimise the decision-making process (i.e. inferring new preferences from previously known preferences). We then analyse the affects of preference semantics on comparative preference statements w.r.t. these postulates, seeking for properties that could characterise their behaviour (§3.3).

3.1 From Preference Models to Comparative Preference Statements and Back

From our brief look at comparative preference statements in Chapter 1, we know that these are a means of compactly representing preferences, and are described as logical formulae using conditional logics. They can be interpreted in terms of the outcomes that satisfy them to compute a preference relation (the preference model) upon the entire set of outcomes. This can be done using different means of completion, by following what are formally known as different preference semantics. While we intimated this passage from preference models to compact representation and back, we did not provide any details, or discuss the technical issues behind. We shall do so now.

3.1.1 Comparative Preference Statements

Let us recall from chapter 1 that the study of preference modelling provided a mathematical basis to describe preferences. We saw that the preference model is a preference relation, or a preference order, which mirrors the definition and properties of the binary relations in order theory.

To describe preference models in a given context, we defined a formal language $\mathcal{L}$, which established the elements to describe a set Ω of outcomes ω, upon which a preference order can be defined. In this context, the preference model becomes the ordered set Ω.

Now, comparative preference statements are defined using conditional logic, and provide a compact description of preferences. They do not explicitly describe the ordered set Ω. What then are the technical steps that lead from comparative preference statements to the preference model Ω?

Let us begin with the construction of comparative preference statements in $\mathcal{L}$.

The compact representation of preferences was defined to cater to the way individuals express their preferences. In particular, one can find that they often, implicitly or explicitly, refer to qualitative comparative preference statements of the form "prefer α to β". Handling such a preference statement is easy when both α and β refer to an outcome, e.g., "I prefer coffee to tea at breakfast". On the other hand, when they refer to sets of outcomes, several complications can arise; particularly when there exist outcomes that belong to both sets.

To elucidate, let us call upon our previous example of choosing a university for further studies. The preference statement "prefer a university in London to a university ranked amongst the top 20%" entails the comparison of two sets of outcomes, Σ_1 and Σ_2, asserting that the universities belonging to Σ_1 are preferred to those in Σ_2. Specifically,

$$\begin{aligned} \Sigma_1 &= \{\text{All universities situated in London}\} \\ \text{is preferred to } \Sigma_2 &= \{\text{All universities ranked amongst the top 20\%}\}. \end{aligned}$$

If there exists a university in London ranked amongst the top 20%, it would, by definition, belong to Σ_1 and to Σ_2. In such a situation we would be faced with a preference of the form α is preferred to α, which is of no use.

To prevent such an occurrence, Halldén (1957) and Von Wright (1963) interpret the statement "prefer α to β" as a choice problem between $\alpha \wedge \neg\beta$-outcomes and $\neg\alpha \wedge \beta$-outcomes. We now take "prefer a university in London to a university ranked amongst the top 20%" to mean "prefer a university in London (ranked below the top 20%) to a university (not situated in London) ranked amongst the top 20%". Consequently,

$$\Sigma_1' = \{\text{All universities situated in London and ranked below the top 20\%}\}$$

is preferred to

$$\Sigma_2' = \{\text{All universities not situated in London and ranked amongst the top 20\%}\}.$$

Particular situations where α (resp. β) is not replaced by $\alpha \wedge \neg\beta$ (resp. $\neg\alpha \wedge \beta$) are when either (1) $\alpha \wedge \neg\beta$ (resp. $\neg\alpha \wedge \beta$) is a contradiction or when (2) there is no contradiction, but the outcomes satisfying $\alpha \wedge \neg\beta$ (resp. $\neg\alpha \wedge \beta$) are not feasible. For further details, we refer the reader to Von Wright (1963) and Hansson (2001). We suppose, for simplicity's sake, that both $\alpha \wedge \neg\beta$ and $\neg\alpha \wedge \beta$ are consistent and feasible and therefore represent disjoint sets of items.

One may also wonder whether "prefer a university in London to a university ranked amongst the top 20%" is a preference statement since it compares the values of two different variables, namely location (i.e., London) and rank (i.e., $\geqslant$ top 20%). In truth, it is an importance statement. That is, it is more important for an individual to choose a university that is in London even though it may be ranked below the top 20% than a university that is ranked amongst the top 20% and not situated in London. Therefore universities in London and ranked below the top 20% are preferred to universities ranked amongst the top 20% but not in London. A statement "prefer α to β" is a preference statement when both α and β refer to the values of the same variable e.g. "prefer London to Paris". In either case, whether the statement "prefer α to β" refers to a preference or an importance, the resulting $\alpha \wedge \neg\beta$-outcomes are preferred to $\neg\alpha \wedge \beta$-outcomes. On this account, we do not make a distinction between a preference statement and an importance statement.

Thus, following the conditions of preference satisfaction (Def. 6), we define comparative preference statements as:

Definition 10 (Comparative Preference Statement)**.** Let α and β be two logical formulae built in $\mathcal{L}$. The *comparative preference statement* $\alpha \rhd \beta$ is defined as a preference of $\alpha \wedge \neg\beta$-outcomes over $\beta \wedge \neg\alpha$-outcomes. We say that an outcome satisfies $\alpha \rhd \beta$ iff it satisfies $\alpha \wedge \neg\beta$.

This definition makes the link between compactly described preference statements and the corresponding outcome ordering in Ω. We see that comparative preference statements express a preference between sets of outcomes. Computing a preference order (the model) based on a comparative preference statement therefore requires interpreting a corresponding preference over individual outcomes. This presents several possibilities, and is resolved by defining different preference semantics.

3.1.2 Preference Semantics

Based on the definition of comparative preference statements above, we can say that a given statement $\alpha \rhd \beta$ can be interpreted as $\mathrm{Mod}(\alpha \wedge \neg\beta)$ preferred to $\mathrm{Mod}(\neg\alpha \wedge \beta)$. This presents several possibilities, depending on how rigorously each outcome in $\mathrm{Mod}(\alpha \wedge \neg\beta)$ sets is required to satisfy the preference. Taking up our example of choosing a university, "I like London more than Paris" could either impose that *all* London universities are preferred to *all* Paris universities, or loosen the requirements and allow exceptions to the preference.

Formally, different semantics have been defined for this purpose, using the notion of (un)dominated outcomes, or sets of minimal and maximal outcomes (see definition 5 on page 11). Thus, considering a set Ω ordered by a given preference relation $\succeq$, the semantics define how $\succeq$ satisfies the comparative preference statement $\alpha \rhd \beta$. This establishes the passage back from comparative preference statements to the preference model of an ordered set. These were first introduced in § 1.2.2.1 on page 17; we now define them formally:

Definition 11 (Preference Semantics)**.** Let $\succeq$ be a preference relation. Consider $\alpha \rhd \beta$.

— *Strong Semantics:*
$\succeq$ satisfies $\alpha \triangleright \beta$, denoted by $\succeq \models_{st} \alpha \triangleright \beta$,
iff $\forall \omega \in \min(\alpha \wedge \neg\beta, \succeq)$, $\forall \omega' \in \max(\neg\alpha \wedge \beta, \succeq)$, $\omega \succ \omega'$;

— *Ceteris Paribus Semantics:*
$\succeq$ satisfies $\alpha \triangleright \beta$, denoted by $\succeq \models_{cp} \alpha \triangleright \beta$,
iff $\forall \omega \in \min(\alpha \wedge \neg\beta, \succeq)$, $\forall \omega' \in \max(\neg\alpha \wedge \beta, \succeq)$, $\omega \succ \omega'$,

provided the two outcomes have the same valuation over variables not appearing in $\alpha \wedge \neg\beta$ and $\neg\alpha \wedge \beta$ [1];

— *Optimistic Semantics:*
$\succeq$ satisfies $\alpha \triangleright \beta$, denoted by $\succeq \models_{opt} \alpha \triangleright \beta$,
iff $\forall \omega \in \max(\alpha \wedge \neg\beta, \succeq)$, $\forall \omega' \in \max(\neg\alpha \wedge \beta, \succeq)$, $\omega \succ \omega'$;

— *Pessimistic Semantics:*
$\succeq$ satisfies $\alpha \triangleright \beta$, denoted by $\succeq \models_{pes} \alpha \triangleright \beta$,
iff $\forall \omega \in \min(\alpha \wedge \neg\beta, \succeq)$, $\forall \omega' \in \min(\neg\alpha \wedge \beta, \succeq)$, $\omega \succ \omega'$;

— *Opportunistic Semantics:*
$\succeq$ satisfies $\alpha \triangleright \beta$, denoted by $\succeq \models_{opp} \alpha \triangleright \beta$,
iff $\forall \omega \in \max(\alpha \wedge \neg\beta, \succeq)$, $\forall \omega' \in \min(\neg\alpha \wedge \beta, \succeq)$, $\omega \succ \omega'$.

This definition can be reformulated on the basis of how $\mathrm{Mod}(\alpha \wedge \neg\beta)$ is compared to $\mathrm{Mod}(\neg\alpha \wedge \beta)$, to offer a better understanding of the principles underpinning the semantics:

Definition 11 (bis). Let $\succeq$ be a preference relation and $\alpha \triangleright \beta$ be a comparative preference statement.

— $\succeq \models_{st} \alpha \triangleright \beta$ iff $\forall \omega \in \mathrm{Mod}(\alpha \wedge \neg\beta)$, $\forall \omega' \in \mathrm{Mod}(\neg\alpha \wedge \beta)$, $\omega \succ \omega'$.
— $\succeq \models_{cp} \alpha \triangleright \beta$ iff $\forall \omega \in \mathrm{Mod}(\alpha \wedge \neg\beta)$, $\forall \omega' \in \mathrm{Mod}(\neg\alpha \wedge \beta)$, $\omega \succ \omega'$ provided the two outcomes have the same valuation over variables not appearing in $\alpha \wedge \neg\beta$ and $\neg\alpha \wedge \beta$.
— $\succeq \models_{opt} \alpha \triangleright \beta$ iff $\exists \omega \in \mathrm{Mod}(\alpha \wedge \neg\beta)$, $\forall \omega' \in \mathrm{Mod}(\neg\alpha \wedge \beta)$, $\omega \succ \omega'$.
— $\succeq \models_{pes} \alpha \triangleright \beta$ iff $\exists \omega' \in \mathrm{Mod}(\neg\alpha \wedge \beta)$, $\forall \omega \in \mathrm{Mod}(\alpha \wedge \neg\beta)$, $\omega \succ \omega'$.
— $\succeq \models_{opp} \alpha \triangleright \beta$ iff $\exists \omega \in \mathrm{Mod}(\alpha \wedge \neg\beta)$, $\exists \omega' \in \mathrm{Mod}(\neg\alpha \wedge \beta)$, $\omega \succ \omega'$.

The index of $\models$ (i.e., st, cp, opt, pes, opp) reflects the semantics associated with the comparative preference statement $\alpha \triangleright \beta$. When there is no ambiguity, we shall abuse notation and write $\succeq$ satisfies $\alpha \triangleright_S \beta$ (with $S \in \{st, cp, opt, pes, opp\}$) to mean that $\succeq \models_S \alpha \triangleright \beta$. We also use the symbol $\alpha \triangleright_S \beta$ to say that $\alpha \triangleright \beta$ is interpreted following the corresponding semantics, or that $\alpha \triangleright_S \beta$ is an S-preference. We also note:

1. This is a commonly used interpretation of the semantics.

Definition 12 (Preference Set)**.** A set of S-preferences for $S \in \{st, cp, opt, pes, opp\}$, is defined as $\mathcal{P}_S = \{\alpha \rhd_S \beta\}$. A *preference set*, in general, is denoted by $\mathcal{P}_\rhd$ when it contains preferences associated with several semantics. Thus, $\mathcal{P}_\rhd = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes} \uplus \mathcal{P}_{opp}$.

To formalise the correspondence between preference statements and preference relations, we say that:

Definition 13 (Preference Set Consistency)**.** A preference relation $\succeq$ is a *model* of $\mathcal{P}_\rhd$ if and only if $\forall \alpha \rhd_S \beta \in \mathcal{P}_\rhd, \succeq \models_S \alpha \rhd \beta$. A preference set is *consistent* if and only if it has a model.

Definition 11(bis) reveals that the five semantics express more or less requirements on the way $\alpha \wedge \neg\beta$-outcomes and $\neg\alpha \wedge \beta$-outcomes are rank-ordered. Strong semantics impose the most requirements, as all outcomes must be ordered according to their satisfaction of the preference $\alpha \rhd \beta$, regardless of all other variables. This means that all $\alpha \wedge \neg\beta$-outcomes are necessarily ordered with a higher preference to all $\neg\alpha \wedge \beta$-outcomes. Indeed, it can easily be verified that with $\alpha \rhd_{st} \beta$ we automatically have $\alpha \rhd_{cp} \beta$, $\alpha \rhd_{opt} \beta$, $\alpha \rhd_{pes} \beta$ and $\alpha \rhd_{opp} \beta$ since all the other semantics weaken the requirements of strong semantics (Kaci, 2011, §3.4.7 on p.48).

To continue with our example, in choosing a university for further studies according to the variables 'rank' and 'location', a strong preference for 'London' over 'Paris' would result in ordering all London universities above Paris universities, regardless of their rank; and this rank-ordering would remain valid whatever be the preference semantics associated with $\alpha \rhd \beta$.

Strong semantics has been criticised in the literature since it may lead to cyclic (i.e. contradictory) preferences when several preference statements are considered. Considering for example:

$$p_1 = \textit{London} \rhd_{st} \textit{Paris} \quad \text{and} \quad p_2 = \in\!\textit{Top 20\%} \rhd_{st} \notin\!\textit{Top 20\%},$$

there is no acyclic preference relation satisfying both statements since top ranked universities situated in Paris would violate p_1, and universities in London ranked below the top 20% would violate p_2. A cyclic preference relation in this case would be:

$$\text{London} + \notin \text{Top 20\%} \succ \text{Paris} + \in \text{Top 20\%} \qquad \text{w.r.t. } p_1,$$

and

$$\text{Paris} + \in \text{Top 20\%} \succ \text{London} + \notin \text{Top 20\%} \qquad \text{w.r.t. } p_2.$$

Ceteris paribus semantics are a good alternative in such situations as they slacken these requirements by adding a further constraint upon the variables that are not concerned in the preference $\alpha \rhd \beta$. This reduces the number of $\alpha \wedge \neg\beta$-outcomes and $\neg\alpha \wedge \beta$-outcomes compared. In this way, considering this time:

$$p_1 = \textit{London} \rhd_{cp} \textit{Paris} \quad \text{and} \quad p_2 = \in\!\textit{Top 20\%} \rhd_{cp} \notin\!\textit{Top 20\%},$$

we have an acyclic rank-ordering satisfying both preferences, when London universities are above Paris universities only when they have the same rank. In other words, we could have:

$$\begin{array}{ll} & \text{all top ranking } (\in \text{ top } 20\%) \text{ London universities} \\ \succ & \text{all top ranking Paris universities} \\ \succ & \text{all low ranking } (\notin \text{ top } 20\%) \text{ London universities} \\ \succ & \text{all low ranking Paris universities,} \end{array}$$

which is definitely more desirable.

In optimistic semantics we have a left-hand weakening of strong semantics. Here left-hand refers to α in the statement $\alpha \rhd \beta$ and by extension to $\alpha \wedge \neg\beta$-outcomes as opposed to $\neg\alpha \wedge \beta$-outcomes. Thus left-hand weakening implies relaxing requirements for $\alpha \wedge \neg\beta$-outcomes. Instead of insisting that any $\alpha \wedge \neg\beta$-outcome is preferred to any $\neg\alpha \wedge \beta$-outcome (as in strong semantics), with optimistic semantics one needs merely to have at least one $\alpha \wedge \neg\beta$-outcome preferred to any $\neg\alpha \wedge \beta$-outcome. This reflects an increase in flexibility concerning the outcome(s) which fulfil this requirement. The larger the set of $\alpha \wedge \neg\beta$-outcomes, the more flexible the statement $\alpha \rhd \beta$. Flexibility should be understood as the number of possible preference relations satisfying $\alpha \rhd \beta$. Continuing with our example, since an optimistic semantics requires preferring at least one top ranking London university above all the other universities,

$$p_1 = \textit{London} \rhd_{\text{opt}} \textit{Paris} \quad \text{and} \quad p_2 = \in\!\textit{Top 20\%} \rhd_{\text{opt}} \notin\!\textit{Top 20\%}$$

can be satisfied by the following rank-ordering:

$$\begin{array}{lll} & & \text{all top ranking } (\in \text{ top } 20\%) \text{ London universities} \\ \succ & \text{all top ranking } (\in \text{ top } 20\%) \text{ Paris universities} & \\ & \approx & \text{all low ranking } (\notin \text{ top } 20\%) \text{ London universities} \\ & \approx & \text{all low ranking } (\notin \text{ top } 20\%) \text{ Paris universities.} \end{array}$$

We can see that this rank-ordering is invalid for the examples using strong and ceteris paribus semantics.

Pessimistic semantics is a right-hand weakening of strong semantics. Based on the reasoning we provided for optimistic semantics, we can deduce that it requires that at least one $\neg\alpha \wedge \beta$-outcome should be less preferred to any $\alpha \wedge \neg\beta$-outcome, and that the larger the set of $\neg\alpha \wedge \beta$-outcomes, the more flexible the statement $\alpha \rhd \beta$.

Lastly, opportunistic semantics is both left- and right-hand weakening of strong semantics since it requires that at least one $\alpha \wedge \neg\beta$-outcome should be preferred to at least one $\neg\alpha \wedge \beta$-outcome.

Among the five semantics, ceteris paribus has been given much attention within the research communities of artificial intelligence, philosophy and psychology (Boutilier et al., 2004, Wilson, 2004, Hansson, 2001, Van Benthem et al., 2009, Schiffer, 1991, Earman and Roberts, 1999). Strong, optimistic, pessimistic and opportunistic semantics (in particular the latter three) have not benefitted from the same depth of scrutiny in the preference representation community. They have, however, been studied from an algorithmic point of view. As seen in Pearl (1990), Kaci and van der Torre (2008), Wilson (2004), Benferhat et al. (2002b), given a set of preference statements and a semantics, algorithms have been developed to compute a distinguished preference relation associated with this set. In the next section, we look more closely at algorithms that compute these relations on the basis of the principle of specificity, and follow it up with a postulate based analysis of the behavioural aspects of the five semantics.

3.2 Algorithms to Compute Preference Relations with Comparative Preference Statements

In the previous section, we saw how comparative preference statements can be interpreted to induce a preference relation on a set of outcomes, by defining different semantics to that end. We also indicated in chapter 1 (p.13) that the principle of specificity can be used to handle situations in the presence of defeasible preferences. This leads to computing what we call distinguished preference relations.

We look more closely at this aspect in this section, providing a context and a proper definition for specificity (§3.2.1) and describing algorithms that base themselves on this definition to compute distinguished preference relations (§3.2.2).

3.2.1 Specificity, Non-monotonicity and Distinguished Preference Relations

In our general review of preferences, we pointed out how non-monotonicity is a convenient way of dealing with uncertainty and default knowledge when reasoning with preferences (§ 1.1.2 on page 11). As regards comparative preference statements, the necessity of using this form of reasoning to compute preference relations becomes apparent when looking closely at the expressive power of the different semantics associated with them. In the presence of uncertainty and default knowledge, all the semantics we visited above are not equally suitable to the task. Consider the following example:

Let us suppose that an individual would prefer a Paris university to a London university except if the university offers an optional drama course. This means that we have:

$$p_1 = \mathit{Paris} \rhd \mathit{London} \quad \text{and} \quad p_2 = \mathit{drama} \wedge \mathit{London} \rhd \mathit{drama} \wedge \mathit{Paris}.$$

In associating a semantics to these statements, both strong semantics and ceteris paribus semantics return contradictory (i.e., cyclic) preferences on the outcomes: ***Paris + drama*** is preferred to ***London + drama*** w.r.t. p_1 and ***London + drama*** is preferred to ***Paris + drama*** w.r.t. p_2. This is an undesirable situation because p_1 and p_2 are not contradictory. They simply state that an individual has a default preference for Paris over London but if an optional drama course is offered then they would prefer London.

On the other hand, given that optimistic semantics requires that at least one $\alpha \wedge \neg\beta$-outcome should be preferred to any $\neg\alpha \wedge \beta$-outcome, it leaves room for exceptions. Thus, p_1 and p_2 can be consistently handled together by associating this semantics. Indeed, the preference relation:

$$\begin{array}{lll} \text{Paris + any course other than drama} & & \\ \quad \succ \ \text{London + any course other than drama} & \approx & \text{London + drama} \\ \quad \succ \ \text{Paris + drama} & & \end{array}$$

satisfies both statements w.r.t. optimistic semantics.

Associating the statements with pessimistic semantics works in a dual way w.r.t. optimistic semantics. The following preference relation satisfies the two preference statements w.r.t. pessimistic semantics:

$$\begin{array}{lll} \text{London + drama} & & \\ \quad \succ \ \text{Paris + any course other than drama} & \approx & \text{Paris + drama} \\ \quad \succ \ \text{London + any course other than drama} & & \end{array}$$

Finally, both preference relations above satisfy p_1 and p_2 w.r.t opportunistic semantics. It is the weakest semantics, but that doesn't prevent it from having its own share of uses. We refer the reader to (Van der Torre and Weydert, 2001) where an example shows that a preference relation can be derived using opportunistic semantics, but none of the other semantics.

We can therefore see that beyond the technical device of the five semantics as concerns the selection of at least one or all $\alpha \wedge \neg\beta$-outcomes and $\neg\alpha \wedge \beta$-outcomes, some semantics can be highlighted for their expressive power. Although strong and ceteris paribus semantics are the most natural among the five semantics, they do not leave much room for exceptions, and this makes them unsuitable to reason about defeasible preferences. The workaround for these two semantics comes from using non-monotonic reasoning: the concept of specificity.

Specificity of Preference Statements Continuing with the example above, the preference for Paris over London should be maintained for all universities except when they offer an optional drama course. This makes every university offering an optional drama course an exceptional case which "enforces" the inverted preference. Following defeasible reasoning terminology we say that p_2 is more *specific* than p_1 because the former is true in the context of an optional drama course while the latter is expressed in a more general context: p_2 takes precedence over p_1.

In order to deal with defeasible preferences interpreted using ceteris paribus semantics, Tan and Pearl (1994) rank-order comparative preference statements w.r.t. their specificity. Thus ceteris paribus semantics is first applied to the most specific preferences. Less specific preferences are then considered so long as they do not lead to a contradiction. Therefore we first have *London* + *drama* $\succ$ *Paris* + *drama* since p_2 takes precedence over p_1, and then we have *Paris* + *any course other than drama* $\succ$ *London* + *any course other than drama* considering p_1. Van Benthem et al. (2009) distinguish these as 'normal' situations. That is, p_1 is applied in a normal situation, namely when $\neg \mathtt{drama}$ is true. We can thus say that:

$$p_1' = \neg \mathit{drama} \wedge \mathit{Paris} \rhd \neg \mathit{drama} \wedge \mathit{London}$$
$$\text{with} \quad p_2 = \mathit{drama} \wedge \mathit{London} \rhd \mathit{drama} \wedge \mathit{Paris}.$$

Note however that in both works we need additional information about the specificity between preference statements and normal situations. Using optimistic and pessimistic semantics to deal with defeasible knowledge (Pearl, 1990, Benferhat et al., 2002b), this is not required.

Specificity of Preference Relations Using specificity to rank-order preferences so that specific ones are considered before indicates how the preference statements can be handled consistently. However, since less specific preferences are considered *so long as they do not lead to a contradiction*, this means that *all* the models of the less specific preferences are not valid anymore. Only those that do not lead to a contradiction are valid.

To properly choose the valid models, we must resort to a non-monotonic logic. By applying the notion of specificity to preference relations (and not statements), these models can be selected from the set of all models of the preference statements. We borrow the terminology from Kaci and van der Torre (2008) and define these models as *distinguished* models. Formally, the specificity relation among preference relations is defined as following:

Definition 14 (Specificity)**.** Let $\succeq$ and $\succeq'$ be two total preorders on a set of outcomes Ω, respectively represented by the ordered partitions $(E_1, ..., E_n)$ and $(E_1', ..., E_{n'}')$. We say that $\succeq$ is *less specific than* $\succeq'$, written as $\succeq \sqsubseteq \succeq'$, iff $\forall \omega, \omega' \in \Omega$, if $\omega \in E_i$ and $\omega \in E_j'$ then $i \leqslant j$.

Defined in this way, $\sqsubseteq$ orders total preorders by preserving $\leqslant$ on disjoint equivalence classes. Given a set of preference statements $\mathcal{P}_{\rhd}$, the set of all models of this set can therefore be ordered by $\sqsubseteq$. The *distinguished preference models*, then, are seen as the *least-* and *most-specific* preference relations in this set (when they exist).

The existence and unicity of these models have been studied in several works, where the different semantics have been studied separately (Pearl, 1990, Benferhat et al., 1999, 2001, Benferhat and Kaci, 2001, Benferhat et al., 2002a, Dubois et al., 2004) or together (Kaci and van der Torre, 2008). The results of these works are summarised in Table 3.1.

Distinguished Models	$\mathcal{P}_{st}$	$\mathcal{P}_{cp}$	$\mathcal{P}_{opt}$	$\mathcal{P}_{pes}$	$\mathcal{P}_{opp}$	$\mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt}$	$\mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt}$
least-specific	✓	✓	✓	–	–	✓	–
most-specific	✓	✓	–	✓	–	–	✓

Table 3.1 – Existence and Unicity of Distinguished Models of Preference Sets

Proofs from these studies indicate the use of the technical device of the $\mathcal{MAX}$ and $\mathcal{MIN}$ operators defined below (Def. 15) to show that these distinguished models are unique when they exist.

Definition 15 (Maximum and Minimum of Two Preference Relations). Let $\succeq$ and $\succeq'$ be two total preorders on a set of outcomes Ω, respectively represented by the ordered partitions $(E_1, ..., E_n)$ and $(E'_1, ..., E'_{n'})$ with $n \geqslant n'$. Let $E'_j = \emptyset$ for $n' < j \leqslant n$.

The maximum and minimum of $\succeq$ and $\succeq'$ are respectively computed by defining the $\mathcal{MAX}$ and $\mathcal{MIN}$ operators as follows:

$$\mathcal{MAX}(\succeq, \succeq') = (E''_1, \ldots, E''_{n'}) \text{ where } E''_i = \begin{cases} E_i \cup E'_i & \text{if } i = 1 \\ E_i \cup E'_i - \bigcup_{j=1}^{i-1} E''_j & \text{if } 1 < i < n' \end{cases}$$

$$\mathcal{MIN}(\succeq, \succeq') = (E'''_1, \ldots, E'''_n) \text{ where } E'''_i = \begin{cases} E_i \cup E'_i & \text{if } i = n \\ E_i \cup E'_i - \bigcup_{j=i+1}^{n} E'''_j & \text{if } 0 < i < n \end{cases}$$

For both operators, the empty sets E''_i and E'''_i are removed and the non-empty sets are renumbered in sequence.

The unicity of these distinguished preference relations can also be retrieved from a lattice-theoretic point of view. Given a partially ordered set of total preorders Γ, The $\mathcal{MAX}$ (resp. $\mathcal{MIN}$) operator represents the least upper (resp. greatest lower) bound, or join (resp. meet), or supremum (resp. infimum), of every non-empty finite subset in Γ. This induces an upper or join (resp. lower or meet) semi-lattice on Γ. The join (resp. meet) of Γ is the least-specific (resp. most-specific) preorder in Γ and is thereby ***unique***.

From Table 3.1 we can therefore deduce that the set of models for:
— $\mathcal{P}_{\mathrm{st}}$ and $\mathcal{P}_{\mathrm{cp}}$ is a semi-lattice,
— $\mathcal{P}_{\mathrm{opt}}$ is a join semi-lattice, and
— $\mathcal{P}_{\mathrm{pes}}$ is a meet semi-lattice.

This provides an incentive for the development of algorithms to compute them from a given set of preferences. These are individually defined in the works mentioned above, but can also be found all together in Kaci (2011, p.42-48, 55-61). We complete this section with a discussion of these algorithms.

3.2.2 Algorithms to Compute Distinguished Preference Relations

We now present the algorithms that compute the distinguished preference relations associated with $\mathcal{P}_{\mathrm{st}}, \mathcal{P}_{\mathrm{cp}}, \mathcal{P}_{\mathrm{opt}}, \mathcal{P}$ and $_{\mathrm{pes}}$ and discuss their relevance in computing preference relations from a preference set associated with several semantics $\mathcal{P}_{\triangleright}$.

Each of these algorithms follows a general construction process which is identical, and can be seen as a step by step construction of the distinguished model of a preference set P_S by following Definition 11 (bis). This common construction process is summarised below.

Given the input set P_S, the distinguished model is an ordered partition Ω, obtained by classifying the outcomes in Ω as follows:

1. Compute $\mathrm{Mod}(\alpha \wedge \neg\beta)$ and $\mathrm{Mod}(\neg\alpha \wedge \beta)$ for each $\alpha \triangleright_S \beta$ in P_S. These pairs form a set of constraints $\mathcal{C}$,
2. Construct one class of the ordered partition by determining all the maximal (resp. minimal) as-yet-unclassified outcomes when computing the least-specific (resp. most specific) model,
3. If no outcomes are found, EXIT algorithm (preferences are inconsistent).
4. Update $\mathcal{C}$ w.r.t S semantics to (1) exclude classified outcomes from individual constraints and (2) remove satisfied constraints,
5. Repeat from step 1 till all outcomes are classified.

Step 1 identifies the subsets of Ω which are to be compared to generate a set of constraints. Steps 2 and 3 combined result in the application of Definition 11 (bis) on the outcomes in the set of constraints w.r.t the semantics specified in P_S. Step 4 then updates the set of constraints w.r.t the classified outcomes to proceed to Step 5, by which the remaining unclassified outcomes in Ω can be classified in their turn.

We can observe that Step 1 always generates the same set of constraints, irrespective of the semantics associated with $\mathcal{P}_S$. Initialising this set is therefore a common first step in all algorithms. Formally, we say that given a comparative preference statement p, a *constraint* is an ordered pair $c(p) = (L(p), R(p))$, where $L(p) = \mathrm{Mod}(\alpha \wedge \neg\beta)$ and $R(p) = \mathrm{Mod}(\neg\alpha \wedge \beta)$. Given a set of comparative preference statements $\mathcal{P}_{\triangleright}$, the set of all constraints $c(p)$ induced by each $p \in \mathcal{P}_{\triangleright}$ is defined as a *constraint set* induced by $\mathcal{P}_{\triangleright}$, denoted by $\mathcal{C}(\mathcal{P}_{\triangleright})$.

3.2.2.1 Algorithms for $\mathcal{P}_{st}$

Algorithm 1: Computing the Least-Specific Model of $\mathcal{P}_{st}$

Data: A preference set $\mathcal{P}_{st}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.
Initialise $\mathcal{C}(\mathcal{P}_{st})$;
$i = 0$;
while $\Omega \neq \emptyset$ do
    $i = i + 1$;
    $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{st}), \omega \notin R(p)\}$;
    if $E_i == \emptyset$ then
        Exit Algorithm due to Inconsistent Preferences;
    for $\omega \in E_i$ do
        Remove ω from Ω;
        Remove ω from each $L(p)$ of constraints in $\mathcal{C}(\mathcal{P}_{st})$;
    Remove constraints with $L(p) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{st})$;
return $(E_1, \ldots E_i)$

Algorithm 2: Computing the Most-Specific Model of $\mathcal{P}_{st}$

Data: A preference set $\mathcal{P}_{st}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.
Initialise $\mathcal{C}(\mathcal{P}_{st})$;
$i = 0$;
while $\Omega \neq \emptyset$ do
    $i = i + 1$;
    $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{st}), \omega \notin L(p)\}$;
    if $E_i == \emptyset$ then
        Exit Algorithm due to Inconsistent Preferences;
    for $\omega \in E_i$ do
        Remove ω from Ω;
        Remove ω from each $R(p)$ of constraints in $\mathcal{C}(\mathcal{P}_{st})$;
    Remove constraints with $L(p) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{st})$;
return $(E_i, \ldots E_1)$

We now discuss each algorithm in detail.

Recalling that the set of models for a set of preferences $\mathcal{P}_{st}$ can be characterised as a semi-lattice, we can deduce that it contains *two distinguished models*: the least upper bound, or the least-specific preference relation, and the greatest lower bound, or the most-specific preference relation. Algorithm 1 computes the former, and Algorithm 2 computes the latter.

Looking over the structure of two algorithms, one can identify the general construction process that was described in the beginning of this subsection. Upon closer observation, differences can be seen in lines 5, 10 and 12.

This is due to the manner in which the resulting ordered partition is constructed. To construct the least-specific ordered partition, Algorithm 1 computes the maximal, or most preferred outcomes at each step, and thus the order in which it generates each class is identical to that of the ordered partition to be returned. This is reflected in lines 5, 10 and 12. In lines 5 and 10, the outcomes in $\mathsf{L}(p)$ are indeed the maximal outcomes and line 12 shows that the classes are constructed by order of preference.

On the other hand, to construct the most-specific ordered partition, Algorithm 2 proceeds in the reverse order: at every step it computes the minimal, or least preferred outcomes. The classes are therefore generated in the opposite order, and must be returned last to first. This is reflected in lines 5, 10 and 12.

The fact that these algorithms deal with strong preferences is reflected in the way constraints are updated after generating one class of the partition. Updating the constraints means that (1) individual constraints induced by strong preferences must be updated to exclude already classified outcomes and (2) satisfied constraints must be removed. This is performed in lines 10 and 11.

Recall that by Definition 11 (bis), for a given statement $p = \alpha \rhd \beta$, *all* outcomes in $\mathsf{Mod}(\alpha \wedge \neg\beta)$ have to be preferred to *all* those in $\mathsf{Mod}(\neg\alpha \wedge \beta)$. Thus, in Algorithm 1 (resp. Algorithm 2), the maximal (resp. minimal) outcomes that have already been classified have to be removed from $\mathsf{Mod}(\alpha \wedge \neg\beta)$ (resp $\mathsf{Mod}(\neg\alpha \wedge \beta)$), i.e. $\mathsf{L}(p)$ (resp. $\mathsf{R}(p)$), so that the next set of maximal (resp. minimal) outcomes can be determined from those remaining in $\mathsf{L}(p)$ (resp. $\mathsf{R}(p)$). This is repeated until $\mathsf{L}(p)$ (resp. $\mathsf{R}(p)$) is empty, since it is only then that the preference is satisfied.

3.2.2.2 Algorithms for $\mathcal{P}_{cp}$

The algorithms for $\mathcal{P}_{cp}$ are identical to those for $\mathcal{P}_{st}$ except in line 10: the line where constraints are updated. This is because it is only here that the ceteris paribus condition is applied.

When an outcome ω has been determined as maximal (resp. minimal), it is removed from each $\mathsf{L}(p)$ (resp. $\mathsf{R}(p)$) for the same reason as that for strong preferences. The application of the ceteris paribus clause to this ω means that there is a one-to-one correspondence between it and its "partner" ω' in $R(p)$ (resp. $L(p)$), such that $\omega \succ \omega'$ w.r.t. ceteris paribus. This "partner" must also be removed, since it cannot be compared to any other outcome w.r.t the ceteris paribus clause.

Algorithm 3: Computing the Least-Specific Model of $\mathcal{P}_{cp}$

Data: A preference set $\mathcal{P}_{cp}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.

Initialise $\mathcal{C}(\mathcal{P}_{cp})$;
$i = 0$;
while $\Omega \neq \emptyset$ **do**
  $i = i + 1$;
  $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{cp}), \omega \notin R(p)\}$;
  if $E_i == \emptyset$ **then**
    Exit Algorithm due to Inconsistent Preferences;
  for $\omega \in E_i$ **do**
    Remove ω from Ω;
    Remove ω and ω' respectively from each $L(p)$ and $R(p)$ of constraints in $\mathcal{C}(\mathcal{P}_{cp})$, where $\omega \succ_{cp} \omega'$;
  Remove constraints with $L(p) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{cp})$;
return $(E_1, \ldots E_i)$

Algorithm 4: Computing the Most-Specific Model of $\mathcal{P}_{cp}$

Data: A preference set $\mathcal{P}_{cp}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.

Initialise $\mathcal{C}(\mathcal{P}_{cp})$;
$i = 0$;
while $\Omega \neq \emptyset$ **do**
  $i = i + 1$;
  $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{cp}), \omega \notin L(p)\}$;
  if $E_i == \emptyset$ **then**
    Exit Algorithm due to Inconsistent Preferences;
  for $\omega \in E_i$ **do**
    Remove ω from Ω;
    Remove ω and ω' respectively from each $R(p)$ and $L(p)$ of constraints in $\mathcal{C}(\mathcal{P}_{cp})$, where $\omega' \succ_{cp} \omega$;
  Remove constraints with $L(p) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{cp})$;
return $(E_i, \ldots E_1)$

Determining these "partners" needs to be performed before running the algorithms 3 and 4. This is done by defining a *cp-relation* between the outcomes

$\omega \in L(p)$ and $\omega' \in R(p)$, denoted by $\omega \succ_{cp} \omega'$, when they satisfy a preference statement $\alpha \rhd_{cp} \beta$.

3.2.2.3 Algorithms for $\mathcal{P}_{opt}$ and $\mathcal{P}_{pes}$

Algorithm 5: Computing the Least-Specific Model of $\mathcal{P}_{opt}$

Data: A preference set $\mathcal{P}_{opt}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.
1 Initialise $\mathcal{C}(\mathcal{P}_{opt})$;
2 $i = 0$;
3 **while** $\Omega \neq \emptyset$ **do**
4     $i = i + 1$;
5     $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{opt}), \omega \notin R(p)\}$;
6   **if** $E_i == \emptyset$ **then**
7   Exit Algorithm due to Inconsistent Preferences;
8   **for** $\omega \in E_i$ **do**
9   Remove ω from Ω;
10   Remove constraints with $L(p) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{opt})$;
11 **return** $(E_1, \ldots E_i)$

Algorithm 6: Computing the Most-Specific Model of $\mathcal{P}_{pes}$

Data: A preference set $\mathcal{P}_{pes}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.
1 Initialise $\mathcal{C}(\mathcal{P}_{pes})$;
2 $i = 0$;
3 **while** $\Omega \neq \emptyset$ **do**
4     $i = i + 1$;
5     $E_i = \{\omega \mid \forall c(p) \in \mathcal{C}(\mathcal{P}_{pes}), \omega \notin L(p)\}$;
6   **if** $E_i == \emptyset$ **then**
7   Exit Algorithm due to Inconsistent Preferences;
8   **for** $\omega \in E_i$ **do**
9   Remove ω from Ω;
10   Remove constraints with $R(p) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{pes})$;
11 **return** $(E_i, \ldots E_1)$

Recalling that the set of models for $\mathcal{P}_{opt}$ can be characterised as a join semi-lattice, we know that there exists only one distinguished model for this set: the least-specific preorder. Algorithm 5, the algorithm for $\mathcal{P}_{opt}$, is therefore structurally similar to the algorithms 1 and 3 which also compute least-specific preorders. It is, in fact, absolutely identical to them after having omitted line 10. This is because it differs only in the update of constraints, since that is where the semantics of the preferences comes in play.

Considering optimistic semantics for a given preference, as soon as a maximal outcome has been determined, the preference is satisfied. Thus, updating the constraint set only means removing satisfied preferences. This is why line 10 from algorithms 1 and 3 is omitted.

Similarly, Algorithm 6 which is the algorithm for $\mathcal{P}_{pes}$, is identical to the algorithms 2 and 4, after having omitted line 10.

3.2.2.4 Algorithms for $\mathcal{P}_{\triangleright}$

Finally, considering the possibility of having $\mathcal{P}_{\triangleright}$ containing preferences associated with several different semantics, Kaci and van der Torre (2008) prove that:

- The least-specific model of $\mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt}$ is unique.
- The most-specific model of $\mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{pes}$ is unique.

Note that they also distinguish between strict and non-strict preferences associated with strong, ceteris paribus, optimistic and pessimistic semantics and include these in the algorithms. We shall use the following notation to distinguish strict and non-strict preference statements:

- Strict S-preference: $p^{>}_{S}$ (for $S \in \{st, cp, opt, pes\}$),
- Non-strict S-preference: $p^{\geqslant}_{S}$ (for $S \in \{st, cp, opt, pes\}$).

They then propose two algorithms (Algorithms 7 and 8 below) to compute these respectively. Examining lines 20-35 in these algorithms, we can see that they include the update steps from each of the algorithms presented above. We can therefore see that algorithm 7 (resp. algorithm 8) generalises and consequently captures all the algorithms computing least-specific (resp. most-specific) models.

In our decision support system based on comparative preference statements, which we propose in the following chapter, we consider the possibility of having $\mathcal{P}_{\triangleright}$ containing preferences associated with several different semantics, but we do not use these two algorithms to compute recommendations. We propose a method that allows us to use the two simplest algorithms presented here: Algorithms 5 and 6. We will detail this in the following chapter.

Now, having seen how distinguished models can be computed from a set of preferences, we dedicate the rest of this chapter to a behavioural analysis of the preference semantics using existing postulates from preference logics.

Algorithm 7: Computing the Least-Specific Model of $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt}$

Data: A preference set $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.

Initialise $\mathcal{C}(\mathcal{P}_{\triangleright})$;
$i = 0$;
while $\Omega \neq \emptyset$ do
  $i = i + 1$, $b == \mathtt{True}$;
  /** classify ω about strict preferences **/
  $E_i = \{\omega \mid \forall c(p_S^{\geqslant}) \in \mathcal{C}(\mathcal{P}_{\triangleright}), \omega \notin R(p_S^{\geqslant})\}$;
  while $b = \mathtt{True}$ do
    $b = \mathtt{False}$;
    for *each* $c(p_{st}^{\geqslant}), c(p_{cp}^{\geqslant}), c(p_{opt}^{\geqslant}) \in \mathcal{C}(\mathcal{P}_{\triangleright})$ do
      /** update E_i w.r.t. non-strict preferences **/
      if $(L(p_{st}^{\geqslant}) \nsubseteq E_i$ *and* $R(p_{st}^{\geqslant}) \cap E_i \neq \emptyset)$ *or*
      $(L(p_{cp}^{\geqslant}) \cap E_i == \emptyset$ *and* $R(p_{cp}^{\geqslant}) \cap E_i \neq \emptyset)$ *or*
      $(L(p_{opt}^{\geqslant}) \cap E_i == \emptyset$ *and* $R(p_{opt}^{\geqslant}) \cap E_i \neq \emptyset)$ then
        $E_i = E_i \setminus R(p_{st}^{\geqslant}) \cup R(p_{cp}^{\geqslant}) \cup R(p_{opt}^{\geqslant})\}$;
        $b = \mathtt{True}$;
  if $E_i == \emptyset$ then
    Exit Algorithm due to Inconsistent Preferences;
  for $\omega \in E_i$ do
    Remove ω from Ω;
    /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ w.r.t strong preferences (strict and non-strict) **/
    Remove ω from each $L(p_{st}^{>})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$;
    Remove ω from each $L(p_{st}^{\geqslant})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$;
    /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ w.r.t cp preferences (strict and non-strict) **/
    Remove ω and ω' respectively from each $L(p_{cp}^{>})$ and $R(p_{cp}^{>})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$, where $\omega \succ_{cp} \omega'$;
    Remove ω and ω' respectively from each $L(p_{cp}^{\geqslant})$ and $R(p_{cp}^{\geqslant})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$, where $\omega \succ_{cp} \omega'$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied optimistic preferences (strict and non-strict) **/
  Remove $c(p_{opt}^{>})$ with $L(p_{opt}^{>}) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{opt}^{\geqslant})$ with $L(p_{opt}^{\geqslant}) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied strong preferences (strict and non-strict) **/
  Remove $c(p_{st}^{>})$ with $L(p_{st}^{>}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{st}^{\geqslant})$ with $L(p_{st}^{\geqslant}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied cp preferences (strict and non-strict) **/
  Remove $c(p_{cp}^{>})$ with $L(p_{cp}^{>}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{cp}^{\geqslant})$ with $L(p_{cp}^{\geqslant}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
return $(E_1, \ldots E_i)$

Algorithm 8: Computing the Most-Specific Model of $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{pes}$

Data: A preference set $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{pes}$ on a set of outcomes Ω.
Result: An ordered partition of Ω, written as $(E_1, \ldots, E_n)$.

Initialise $\mathcal{C}(\mathcal{P}_{\triangleright})$;
$i = 0$;
while $\Omega \neq \emptyset$ do
  $i = i + 1$, $b == \texttt{True}$;
  /** classify ω about strict preferences **/
  $E_i = \{\omega \mid \forall c(p_S^{>}) \in \mathcal{C}(\mathcal{P}_{\triangleright}), \omega \notin L(p_S^{>})\}$;
  while $b = \texttt{True}$ do
    $b = \texttt{False}$;
    for *each* $c(p_{st}^{\geqslant}), c(p_{cp}^{\geqslant}), c(p_{opt}^{\geqslant}) \in \mathcal{C}(\mathcal{P}_{\triangleright})$ do
      /** update E_i w.r.t. non-strict preferences **/
      if $(L(p_{st}^{\geqslant}) \cap E_i \neq \emptyset$ ***and*** $R(p_{st}^{\geqslant}) \not\subseteq E_i \neq \emptyset)$ ***or***
      $(L(p_{cp}^{\geqslant}) \cap E_i \neq \emptyset$ ***and*** $R(p_{cp}^{\geqslant}) \cap E_i == \emptyset)$ ***or***
      $(L(p_{opt}^{\geqslant}) \cap E_i \neq \emptyset$ ***and*** $R(p_{opt}^{\geqslant}) \cap E_i == \emptyset)$ then
        $E_i = E_i \setminus L(p_{st}^{\geqslant}) \cup L(p_{cp}^{\geqslant}) \cup L(p_{opt}^{\geqslant})\}$;
        $b = \texttt{True}$;
  if $E_i == \emptyset$ then
    Exit Algorithm due to Inconsistent Preferences;
  for $\omega \in E_i$ do
    Remove ω from Ω;
    /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ w.r.t strong preferences (strict and non-strict) **/
    Remove ω from each $R(p_{st}^{>})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$;
    Remove ω from each $R(p_{st}^{\geqslant})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$;
    /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ w.r.t cp preferences (strict and non-strict) **/
    Remove ω and ω' respectively from each $R(p_{cp}^{>})$ and $L(p_{cp}^{>})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$, where $\omega' \succ_{cp} \omega$;
    Remove ω and ω' respectively from each $L(p_{cp}^{\geqslant})$ and $R(p_{cp}^{\geqslant})$ of constraints in $\mathcal{C}(\mathcal{P}_{\triangleright})$, where $\omega' \succ_{cp} \omega$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied optimistic preferences (strict and non-strict) **/
  Remove $c(p_{opt}^{>})$ with $R(p_{opt}^{>}) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{opt}^{\geqslant})$ with $R(p_{opt}^{\geqslant}) \cap E_i \neq \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied strong preferences (strict and non-strict) **/
  Remove $c(p_{st}^{>})$ with $R(p_{st}^{>}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{st}^{\geqslant})$ with $R(p_{st}^{\geqslant}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  /** update $\mathcal{C}(\mathcal{P}_{\triangleright})$ by removing satisfied cp preferences (strict and non-strict) **/
  Remove $c(p_{cp}^{>})$ with $R(p_{cp}^{>}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
  Remove $c(p_{cp}^{\geqslant})$ with $R(p_{cp}^{\geqslant}) = \emptyset$ from $\mathcal{C}(\mathcal{P}_{\triangleright})$;
return $(E_i, \ldots E_1)$

3.3 Postulate-Based Analysis of Preference Semantics

In preference logics, as in any logic, a first basis for mathematical reasoning is the establishment of axioms, or postulates. This has indeed been done, as we saw in § 1.1.2 on page 11. We now take up some of these to analyse the preference semantics presented in this chapter, with the intent of bridging the gap between the intuition behind them and the theoretical results obtained. We do mention, however, that the set of postulates we study is not a characterisation of the system of comparative preference statements with associated semantics: it is merely a set of postulates that would help us better understand certain behavioural aspects of preference semantics.

3.3.1 The Postulates

As already seen in Section 2.12, comparative preference statements are one of the many ways of representing preferences using preference logics. Their utility has been further enhanced with the help of preference semantics. In keeping with the aim of formalising the behaviour of these semantics, we select a set of postulates that are in accordance with the intuition behind these semantics. Incidentally, these postulates also show how comparative preference statements could be inferred from other given comparative preference statements.

Recall that a comparative preference statement $\alpha \triangleright \beta$ leads to the comparison of two sets, namely that of $\alpha \wedge \neg\beta$-outcomes and that of $\neg\alpha \wedge \beta$-outcomes. Each semantics then selects at least one or all $\alpha \wedge \neg\beta$- and $\neg\alpha \wedge \beta$-outcomes. For example $\succeq \models_{opt} \alpha \triangleright \beta$ signifies that at least one $\alpha \wedge \neg\beta$-outcome is preferred w.r.t. $\succeq$ to any $\neg\alpha \wedge \beta$-outcome. Additionally, if we have the preference statement $\alpha' \triangleright \beta'$ such that $Mod(\alpha \wedge \neg\beta) \subset Mod(\alpha' \wedge \neg\beta')$ and $Mod(\neg\alpha' \wedge \beta') \subset Mod(\neg\alpha \wedge \beta)$ then we can ensure that $\succeq \models_{opt} \alpha' \triangleright \beta'$. This means that optimistic semantics is tolerant for expanding the set of $\alpha \wedge \neg\beta$-outcomes and reducing the set of $\neg\alpha \wedge \beta$-outcomes. Formally, we define tolerance for expansion/reduction as:

Definition 16 (Expansion/Reduction Tolerance)**.** Let $\succeq$ be a preference relation and $\alpha \triangleright \beta$ be a comparative preference statement. Let $x, y \in \{\exists, \forall\}$.

— A semantics is left- (resp. right-) expansion tolerant iff $\forall \succeq$, if $\succeq \models_S \alpha \triangleright \beta$ then $\succeq \models_S \alpha' \triangleright \beta'$ with $Mod(\alpha \wedge \neg\beta) \subset Mod(\alpha' \wedge \neg\beta')$ (resp. $Mod(\neg\alpha \wedge \beta) \subset Mod(\neg\alpha' \wedge \beta')$).

— A semantics is left- (resp. right-) reduction tolerant iff $\forall \succeq$, if $\succeq \models_S \alpha \triangleright \beta$ then $\succeq \models_S \alpha' \triangleright \beta'$ with $Mod(\alpha' \wedge \neg\beta') \subset Mod(\alpha \wedge \neg\beta)$ (resp.$Mod(\neg\alpha' \wedge \beta') \subset Mod(\neg\alpha \wedge \beta)$).

It is worth noticing that the construction of $\alpha' \rhd \beta'$ is not an end in itself. Our purpose is to construct such a statement in a way that coincides with the intuition behind and serves for real applications. For example given two preference statements $\alpha \rhd \gamma$ and $\alpha \rhd \beta$, one would intuitively expect that $\alpha \rhd \beta \vee \gamma$ and/or $\alpha \rhd \beta \wedge \gamma$ holds. Having constructed $\alpha' \rhd \beta'$ we would be able to check whether the semantics validate this intuition or not. A typical application of such inferences is recommender systems when, based on previous preferences of a user, we try to refine them by inferring new preferences. In addition to postulates related to reduction and expansion principles, we also consider postulates related to coherence and syntax independence. We first list the postulates, attributed to (Van Benthem et al., 2009, Kraus et al., 1990, Barberà et al., 2004, Freund, 2004).

P1:	Coherence	if $\alpha \rhd \beta$ then $\mathtt{not}(\beta \rhd \alpha)$
P2:	Syntax Independence	if $\alpha \equiv \alpha'$ and $\alpha \rhd \beta$ then $\alpha' \rhd \beta$ if $\beta \equiv \beta'$ and $\alpha \rhd \beta$ then $\alpha \rhd \beta'$
P3:	Left Composition	if $\alpha \rhd \gamma$ and $\beta \rhd \gamma$ then $\alpha \vee \beta \rhd \gamma$
P4:	Left Decomposition	if $\alpha \vee \beta \rhd \gamma$ then ($\alpha \rhd \gamma$ and $\beta \rhd \gamma$)
P5:	Right Composition	if $\alpha \rhd \beta$ and $\alpha \rhd \gamma$ then $\alpha \rhd \beta \vee \gamma$
P6:	Right Decomposition	if $\alpha \rhd \beta \vee \gamma$ then ($\alpha \rhd \beta$ and $\alpha \rhd \gamma$)
P7:	Preference Independence	if $\alpha \rhd \beta$ then $\alpha \vee \gamma \rhd \beta \vee \gamma$
P8:	Left Weakening	if $\mathtt{Mod}(\alpha') \subset \mathtt{Mod}(\alpha)$ and $\alpha \rhd \beta$ then $\alpha' \rhd \beta$
P9:	Right Weakening	if $\mathtt{Mod}(\beta') \subset \mathtt{Mod}(\beta)$ and $\alpha \rhd \beta$ then $\alpha \rhd \beta'$

P1 is fairly intuitive. It says that if an individual expresses a strict preference for a statement against another statement then (s)he does not strictly prefer the latter to the former. P2 expresses a syntax independence w.r.t. both α and β. P3 and P5 express the composition of preferred formulae or less preferred ones. At first sight, P4 may appear unnatural because it begins with $\alpha \vee \beta \rhd \gamma$ and concludes with $\alpha \rhd \gamma$ ***and*** $\beta \rhd \gamma$ (and not $\alpha \rhd \gamma$ ***or*** $\beta \rhd \gamma$). Simply going back to the basic interpretation of $\alpha \rhd \beta$ will help understand why. Since $\alpha \vee \beta \rhd \gamma$ leads us to prefer $(\alpha \vee \beta) \wedge \neg\gamma$-outcomes, by the distributive property of $\wedge$ over $\vee$, we also prefer $\alpha \wedge \neg\gamma$-outcomes ***and*** $\beta \wedge \neg\gamma$-outcomes taken separately. Nevertheless, inferring $\alpha \rhd \gamma$ ***or*** $\beta \rhd \gamma$ is also meaningful, and is captured by P8 since $\mathtt{Mod}(\alpha) \subset \mathtt{Mod}(\alpha \vee \beta)$ (Replacing α and β in P8 respectively by $\alpha \vee \beta$

and γ, we can say that $\alpha \triangleright \gamma$ *or* $\beta \triangleright \gamma$ depending on the value of α'). A similar reasoning is drawn in P6.

One may now say that since P8 (resp. P9) logically implies P4 (resp. P6), it is unnecessary to keep both postulates. We have chosen to keep both for our analysis because the subtle distinction between *and* and *or* allows for two distinct cognitive interpretations for the user. In the case of P4 we decompose a left-hand disjunction into a conjunction. A resulting interpretation could be "if I prefer *London* or *Paris* to *Berlin* then I prefer (1) *London* to *Berlin* and (2) *Paris* to *Berlin*". On the other hand, P8 alludes to a simple preference "prefer α to β" and checks whether γ, a logical consequence of α, is also preferred to β. An interpretation of P8 could be "if I prefer *UK* to *Paris*, then I also prefer *London* to *Paris* since *London* implies *UK*. Therefore even if there is a 'logical' implication between the two postulates, it is important to keep both of them.

P7 expresses that if α is preferred to β then the preference holds between two statements that extend them with the same formula. P8 says that if α is preferred to β then a subset of α in terms of outcomes is still preferred to β. P9 applies the same principle to less preferred formulae.

All the postulates except P1 and P2 refer to expansion and/or reduction principles. In the following paragraphs we show how they are involved in postulates P3-P9.

P3 and P4: These two are reciprocal postulates. Given $\alpha \triangleright \gamma$ and $\beta \triangleright \gamma$ we examine how (using which principle) $\alpha \vee \beta \triangleright \gamma$ holds for P3 and vice versa for P4. On the one hand the statements $\alpha \triangleright \gamma$ and $\beta \triangleright \gamma$ lead us to compare $Mod(\alpha \wedge \neg\gamma)$ with $Mod(\neg\alpha \wedge \gamma)$ and $Mod(\beta \wedge \neg\gamma)$ with $Mod(\neg\beta \wedge \gamma)$. On the other hand, $\alpha \vee \beta \triangleright \gamma$ compares $Mod((\alpha \vee \beta) \wedge \neg\gamma)$ with $Mod(\neg\alpha \wedge \neg\beta \wedge \gamma)$. Considering the possible inclusions between the sets mentioned above, we can conclude that:

— $Mod(\alpha \wedge \neg\gamma) \subset Mod((\alpha \vee \beta) \wedge \neg\gamma)$ and $Mod(\beta \wedge \neg\gamma) \subset Mod((\alpha \vee \beta) \wedge \neg\gamma)$ simultaneously confirms (1) the left-expansion of $\alpha \triangleright \gamma$ and $\beta \triangleright \gamma$ in P3 and (2) the left-reduction of $\alpha \vee \beta \triangleright \gamma$ in P4;

— $Mod(\neg\alpha \wedge \neg\beta \wedge \gamma) \subset Mod(\neg\alpha \wedge \gamma)$ and $Mod(\neg\alpha \wedge \neg\beta \wedge \gamma) \subset Mod(\neg\beta \wedge \gamma)$ simultaneously confirms the right-reduction of $\alpha \triangleright \gamma$ and $\beta \triangleright \gamma$ in P3 and (2) the right-expansion of $\alpha \vee \beta \triangleright \gamma$ in P4.

Thus P3 is left-expansion and right-reduction tolerant and P4 is left-reduction and right-expansion tolerant.

P5 and P6: The proof follows in a similar vein for these two reciprocal postulates. Given $\alpha \triangleright \beta$ and $\alpha \triangleright \gamma$ we examine how $\alpha \triangleright \beta \vee \gamma$ holds for P5 and vice versa for P6. In this case the sets of outcomes compared are, on the one hand, $Mod(\alpha \wedge \neg\beta)$ with $Mod(\neg\alpha \wedge \beta)$ and $Mod(\alpha \wedge \neg\gamma)$ with $Mod(\neg\alpha \wedge \gamma)$; while on the other, $Mod(\alpha \wedge \neg(\beta \vee \gamma)) = Mod(\alpha \wedge \neg\beta \wedge \neg\gamma)$ with $Mod(\neg\alpha \wedge (\beta \vee \gamma))$. We can conclude that:

— $\mathrm{Mod}(\neg\alpha \wedge \beta) \subset \mathrm{Mod}(\neg\alpha \wedge (\beta \vee \gamma))$ and $\mathrm{Mod}(\neg\alpha \wedge \gamma) \subset \mathrm{Mod}(\neg\alpha \wedge (\beta \vee \gamma))$ confirms (1) the right-expansion of $\alpha \triangleright \beta$ and $\alpha \triangleright \gamma$ in P5 and (2) the right-reduction of $\alpha \triangleright \beta \vee \gamma$ in P6;

— $\mathrm{Mod}(\alpha \wedge \neg\beta \wedge \neg\gamma) \subset \mathrm{Mod}(\alpha \wedge \neg\beta)$ and $\mathrm{Mod}(\alpha \wedge \neg\beta \wedge \neg\gamma) \subset \mathrm{Mod}(\alpha \wedge \neg\gamma)$ confirms (1) the left-reduction of $\alpha \triangleright \beta$ and $\alpha \triangleright \gamma$ in P5 and (2) the left-expansion of $\alpha \triangleright \beta \vee \gamma$ in P6.

Thus P5 is left-reduction and right-expansion tolerant and P6 is left-expansion and right-reduction tolerant.

P7: Given $\alpha \triangleright \beta$ we examine how $\alpha \vee \gamma \triangleright \beta \vee \gamma$ holds. On the one hand we compare $\mathrm{Mod}(\alpha \wedge \neg\beta)$ with $\mathrm{Mod}(\neg\alpha \wedge \beta)$ and on the other, $\mathrm{Mod}((\alpha \vee \gamma) \wedge \neg(\beta \vee \gamma)) = \mathrm{Mod}(\alpha \wedge \neg\beta \wedge \neg\gamma)$ with $\mathrm{Mod}(\neg(\alpha \vee \gamma) \wedge (\beta \vee \gamma)) = \mathrm{Mod}(\neg\alpha \wedge \gamma \wedge \neg\gamma)$. We can conclude that:

— $\mathrm{Mod}(\alpha \wedge \neg\beta \wedge \neg\gamma) \subset \mathrm{Mod}(\alpha \wedge \neg\beta)$ confirms the left-reduction of $\alpha \triangleright \beta$;

— $\mathrm{Mod}(\neg\alpha \wedge \beta \wedge \neg\gamma) \subset \mathrm{Mod}(\neg\alpha \wedge \beta)$ confirms the right-reduction of $\alpha \triangleright \beta$.

Thus P7 is left- and right-reduction tolerant.

P8: Given $\mathrm{Mod}(\alpha') \subset \mathrm{Mod}(\alpha)$ and $\alpha \triangleright \beta$ we examine how $\alpha' \triangleright \beta$ holds. On the one hand we compare $\mathrm{Mod}(\alpha \wedge \neg\beta)$ with $\mathrm{Mod}(\neg\alpha \wedge \beta)$ and on the other, $\mathrm{Mod}(\alpha' \wedge \neg\beta)$ with $\mathrm{Mod}(\neg\alpha' \wedge \beta)$. Since $\mathrm{Mod}(\alpha') \subset \mathrm{Mod}(\alpha)$, we can safely restrict both sets with the added constraint of satisfying $\neg\beta$ and maintain the inclusion. It follows that $\mathrm{Mod}(\alpha' \wedge \neg\beta) \subset \mathrm{Mod}(\alpha \wedge \neg\beta)$ and consequently confirms the left-reduction of $\alpha \triangleright \beta$.

$\mathrm{Mod}(\alpha') \subset \mathrm{Mod}(\alpha)$ also implies that $\mathrm{Mod}(\neg\alpha) \subset \mathrm{Mod}(\neg\alpha')$. Restricting both sets with the added constraint of satisfying β, we have $\mathrm{Mod}(\neg\alpha \wedge \beta) \subset \mathrm{Mod}(\neg\alpha' \wedge \beta)$. This confirms the right-expansion of $\alpha \triangleright \beta$.

Thus P8 is left-reduction and right-expansion tolerant.

P9: Given $\mathrm{Mod}(\beta') \subset \mathrm{Mod}(\beta)$ and $\alpha \triangleright \beta$ we examine how $\alpha \triangleright \beta'$ holds. With a similar reasoning to that of P8, the sets of outcomes compared are $\mathrm{Mod}(\alpha \wedge \neg\beta)$ and $\mathrm{Mod}(\neg\alpha \wedge \beta)$ on the one hand and $\mathrm{Mod}(\alpha \wedge \neg\beta')$ and $\mathrm{Mod}(\neg\alpha \wedge \beta')$ on the other. Since $\mathrm{Mod}(\beta') \subset \mathrm{Mod}(\beta)$, we can safely restrict both sets with the added constraint of satisfying $\neg\alpha$, maintaining the inclusion. It follows that $\mathrm{Mod}(\neg\alpha \wedge \beta') \subset \mathrm{Mod}(\neg\alpha \wedge \beta)$ and consequently confirms the right-reduction of $\alpha \triangleright \beta$.

$\mathrm{Mod}(\beta') \subset \mathrm{Mod}(\beta)$ also implies that $\mathrm{Mod}(\neg\beta) \subset \mathrm{Mod}(\neg\beta')$. Restricting both sets with the added constraint of satisfying α, we have $\mathrm{Mod}(\alpha \wedge \neg\beta) \subset \mathrm{Mod}(\alpha \wedge \neg\beta')$. This confirms the left-expansion of $\alpha \triangleright \beta$.

Thus P9 is left-expansion and right-reduction tolerant.

	Left-expansion	Left-reduction	Right-expansion	Right-reduction
P1	-	-	-	-
P2	-	-	-	-
P3	✓			✓
P4		✓	✓	
P5		✓	✓	
P6	✓			✓
P7		✓		✓
P8		✓	✓	
P9	✓			✓

Table 3.2 – Left/Right expansion/reduction principles involved in the postulates.

We recapitulate this analysis in Table 3.2 which classifies the principles involved in each postulate, as shown above. On first glance, looking at how left- (resp. right-) expansion and left- (resp. right-) reduction are mutually exclusive, we can already state an impossibility result:

> *If there is no semantics which is simultaneously tolerant for left-expansion, left-reduction, right-expansion and right-reduction, then the postulates cannot all be satisfied together.*

With further analysis, we can group those postulates that share the same principles of expansion/reduction and thus provide sufficient conditions to satisfy subsets of postulates. The following proposition summarises these results.

Proposition 1.

— If a given semantics is left-expansion and right-reduction tolerant then it satisfies P3, P6 and P9.

— If a given semantics is left-reduction and right-expansion tolerant then it satisfies P4, P5 and P8.

— If a given semantics is left-reduction and right-reduction tolerant then it satisfies P7.

Stated thus, Proposition 1 offers a general analysis of any semantics (not necessarily one of the five semantics). This is why it works in one direction (if-then) providing sufficient but not necessary conditions.

In the next subsection, we instantiate these results on the five semantics to investigate their behaviour.

3.3.2 Focus on the five semantics

Having established the expansion/reduction principles and seen how they are involved in the nine preference logic postulates selected to better understand preference semantics, we now focus on the five semantics to see how these principles are involved in them. Eventually, shared principles between the semantics and postulates will serve as links making a final correspondence between them. The following proposition gives the tolerance of each of the five semantics w.r.t. reduction/expansion principles.

Proposition 2. Table 3.3 summarises the tolerance of each semantics for left/right expansion/reduction.

	Left-expansion & Right-expansion	Left-reduction & Right-reduction	Left-expansion & Right-reduction	Left-reduction & Right-expansion
Strong	NO	YES	NO	NO
Ceteris Paribus	NO	YES	NO	NO
Optimistic	NO	NO	YES	NO
Pessimistic	NO	NO	NO	YES
Opportunistic	YES	NO	NO	NO

Table 3.3 – Left/Right expansion/reduction tolerance of the semantics.

For each semantics, we provide a proof when a tolerance is verified, and a counter-example when it is not.

Proof. Strong: $\succeq \models_{st} \alpha \rhd \beta$ means that all $\alpha \wedge \neg\beta$-outcomes are preferred w.r.t. $\succeq$ to any $\neg\alpha \wedge \beta$-outcome. Therefore if we are provided with another preference statement $\alpha' \rhd \beta'$ such that $\mathsf{Mod}(\alpha' \wedge \neg\beta') \subset \mathsf{Mod}(\alpha \wedge \neg\beta)$ and $\mathsf{Mod}(\neg\alpha' \wedge \beta') \subset \mathsf{Mod}(\neg\alpha \wedge \beta)$ we can maintain that all $\alpha' \wedge \neg\beta'$-outcomes are preferred w.r.t. $\succeq$ to any $\neg\alpha' \wedge \beta'$-outcome thereby ensuring that $\succeq \models_{st} \alpha' \rhd \beta'$. This proves that strong semantics is tolerant for left- and right-reduction. The following counterexample proves that it is neither tolerant for left-expansion, nor for right-expansion:

> Consider $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking}, \mathsf{Bottom\text{-}ranking}\}$.
> Given *London* $\wedge$ *Top-ranking* $\rhd$ *Paris* $\wedge$ *Top-ranking*,
> the preference relation
>
> London + Top-ranking
> $\succ$ Paris + Top-ranking $\approx$ London + Bottom-ranking
> $\approx$ Paris + Bottom-ranking

satisfies *London* $\wedge$ *Top-ranking* $\triangleright_{st}$ *Paris* $\wedge$ *Top-ranking* but does not satisfy *London* $\triangleright_{st}$*Paris.*

Ceteris Paribus: The reasoning for ceteris paribus semantics remains identical to that for strong semantics since the two semantics differ merely by the clause "the two outcomes have the same valuation over variables not appearing in $\alpha \wedge \neg\beta$ and $\neg\alpha \wedge \beta$". Therefore if we are provided with the preference statement $\alpha' \triangleright \beta'$ such that $\mathsf{Mod}(\alpha' \wedge \neg\beta') \subset \mathsf{Mod}(\alpha \wedge \neg\beta)$ and $\mathsf{Mod}(\neg\alpha' \wedge \beta') \subset \mathsf{Mod}(\neg\alpha \wedge \beta)$ *which retains the ceteris paribus clause*, we can ensure that $\succeq \models_{cp} \alpha' \triangleright \beta'$. This shows that ceteris paribus semantics is also tolerant for left- and right-reduction. To show that it is neither tolerant for left-expansion, nor for right-expansion we use the same counterexample as for strong semantics.

Optimistic: $\succeq \models_{opt} \alpha \triangleright \beta$ means that at least one $\alpha \wedge \neg\beta$-outcome is preferred w.r.t. $\succeq$ to any $\neg\alpha \wedge \beta$-outcome. Therefore if we are provided with another preference statement $\alpha' \triangleright \beta'$ such that $\mathsf{Mod}(\alpha \wedge \neg\beta) \subset \mathsf{Mod}(\alpha' \wedge \neg\beta')$ and $\mathsf{Mod}(\neg\alpha' \wedge \beta') \subset \mathsf{Mod}(\neg\alpha \wedge \beta)$ we still have at least one $\alpha' \wedge \neg\beta'$-outcome which is preferred w.r.t. $\succeq$ to any $\neg\alpha' \wedge \beta'$-outcome thereby ensuring that $\succeq \models_{opt} \alpha' \triangleright \beta'$. This proves that optimistic semantics is tolerant for left-expansion and right-reduction. The following counterexamples respectively prove that it is neither tolerant for left-reduction, nor for right-expansion:

Consider $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking}, \mathsf{Bottom\text{-}ranking}\}$.

1. Given *London* $\triangleright$ *Paris*,

   $$\begin{aligned} &\text{London + Bottom-ranking} \\ &\quad \succ \text{Paris + Top-ranking} \approx \text{London + Top-ranking} \\ &\quad\quad \approx \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies *London* $\triangleright_{opt}$ *Paris* but does not satisfy *London* $\wedge$ *Top-ranking* $\triangleright_{opt}$ *Paris* $\wedge$ *Top-ranking.*
2. Given *London* $\wedge$ *Top-ranking* $\triangleright$ *Paris* $\wedge$ *Bottom-ranking*,

   $$\begin{aligned} &\text{London + Top-ranking} \approx \text{Paris + Top-ranking} \\ &\quad \succ \text{London + Bottom-ranking} \approx \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies *London* $\wedge$ *Top-ranking* $\triangleright_{opt}$ *Paris* $\wedge$ *Bottom-ranking* but does not satisfy *London* $\triangleright_{opt}$ *Paris.*

Pessimistic: $\succeq \models_{pes} \alpha \triangleright \beta$ means that at least one $\neg\alpha \wedge \beta$-outcome should be less preferred w.r.t. $\succeq$ to any $\alpha \wedge \neg\beta$-outcome.Therefore if we are provided with another preference statement $\alpha' \triangleright \beta'$ such that $\mathsf{Mod}(\alpha' \wedge \neg\beta') \subset \mathsf{Mod}(\alpha \wedge \neg\beta)$ and $\mathsf{Mod}(\neg\alpha \wedge \beta) \subset \mathsf{Mod}(\neg\alpha' \wedge \beta')$ we still have at least one $\neg\alpha' \wedge \beta'$-outcome which is less preferred w.r.t. $\succeq$ to any $\alpha' \wedge \neg\beta'$-outcome thereby ensuring that $\succeq \models_{pes} \alpha' \triangleright \beta'$. This proves that pessimistic semantics is tolerant for left-reduction and right-expansion. The following counterexamples respectively prove that it is neither tolerant for left-expansion nor for right-reduction:

Consider $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}, \mathtt{Bottom\text{-}ranking}\}$.

1. Given *London* $\wedge$ *Top-ranking* $\rhd$ *Paris* $\wedge$ *Bottom-ranking*,

$$\begin{aligned} &\text{London + Top-ranking} \\ &\quad \succ \ \text{Paris + Top-ranking} \approx \text{London + Bottom-ranking} \\ &\qquad \approx \ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies *London* $\wedge$ *Top-ranking* $\rhd_{\mathtt{pes}}$ *Paris* $\wedge$ *Bottom-ranking* but does not satisfy *London* $\rhd_{\mathtt{pes}}$ *Paris*.

2. Given *London* $\rhd$ *Paris*,

$$\begin{aligned} &\text{London + Top-ranking} \quad \approx \quad \text{Paris + Top-ranking} \\ &\quad \succ \ \text{London + Bottom-ranking} \\ &\quad \succ \ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies *London* $\rhd_{\mathtt{pes}}$ *Paris* but does not satisfy *London* $\wedge$ *Top-ranking* $\rhd_{\mathtt{pes}}$ *Paris* $\wedge$ *Top-ranking*.

Opportunistic: $\succeq \models_{\mathtt{opp}} \alpha \rhd \beta$ means that at least one $\alpha \wedge \neg\beta$-outcome is preferred w.r.t. $\succeq$ to at least one $\neg\alpha \wedge \beta$-outcome. Therefore if we are provided with another preference statement $\alpha' \rhd \beta'$ such that $\mathtt{Mod}(\alpha \wedge \neg\beta) \subset \mathtt{Mod}(\alpha' \wedge \neg\beta')$ and $\mathtt{Mod}(\neg\alpha \wedge \beta) \subset \mathtt{Mod}(\neg\alpha' \wedge \beta')$ we still have at least one $\alpha' \wedge \neg\beta'$-outcome which is preferred w.r.t. $\succeq$ to at least one $\neg\alpha' \wedge \beta'$-outcome thereby ensuring that $\succeq \models_{\mathtt{opp}} \alpha' \rhd \beta'$. This proves that opportunistic semantics is tolerant for left- and right-expansion. The following counterexample proves that it is neither tolerant for left-reduction nor for right-reduction:

Consider $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}, \mathtt{Bottom\text{-}ranking}\}$.
Given *London* $\rhd$ *Paris*,

$$\begin{aligned} &\text{London + Top-ranking} \quad \approx \quad \text{Paris + Top-ranking} \\ &\quad \succ \ \text{London + Bottom-ranking} \quad \approx \quad \text{Paris + Bottom-ranking} \end{aligned}$$

satisfies *London* $\rhd_{\mathtt{opp}}$ *Paris* but does not satisfy *London* $\wedge$ *Top-ranking* $\rhd_{\mathtt{opp}}$ *Paris* $\wedge$ *Top-ranking*.

$\square$

Given Proposition 1 and Table 3.3, Table 3.4 reports the five semantics and their postulate satisfaction. A satisfaction implies that any preference relation $\succeq$ which satisfies the antecedent of "If" also satisfies its consequence. For example a given semantics satisfies P1 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ then $\succeq$ does not satisfy $\beta \rhd \alpha$. Table 3.4 ensures that if a semantics is tolerant to a reduction/expansion and such a reduction/expansion is involved in a postulate then the semantics satisfies the postulate in question. For example optimistic semantics is left-expansion and right-reduction tolerant. As the latter principles are involved in P3, P6 and P9, optimistic semantics satisfies these postulates. YES that are marked with * do not follow from Proposition 1. We also recall that the satisfaction of P1 and P2 cannot follow from Proposition 1 either since they do not refer to principles of reduction/expansion. We now provide proofs for the postulate satisfaction of the five semantics. We provide a proof when a postulate is satisfied (indicated by YES in Table 3.4) and a counter-example when it is not (indicated by NO in Table 3.4).

Postulates	Strong	Ceteris Paribus	Optimistic	Pessimistic	Opportunistic
P1: Coherence	YES	YES	YES	YES	NO
P2: Syntax Independence	YES	NO	YES	YES	YES
P3: Left Composition	YES*	NO	YES	NO	NO
P4: Let Decomposition	NO	NO	NO	YES	NO
P5: Right Composition	YES*	NO	NO	YES	NO
P6: Right Decomposition	NO	NO	YES	NO	NO
P7: Preference Independence	YES	YES	NO	NO	NO
P8: Left Weakening	NO	NO	NO	YES	NO
P9: Right Weakening	NO	NO	YES	NO	NO

Table 3.4 – Postulate satisfaction.

Proof. P1: A given semantics satisfies P1 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ then $\succeq$ does not satisfy $\beta \rhd \alpha$. If $\succeq \models_S \alpha \rhd \beta$ then we can ensure that (1) for strong, ceteris paribus and optimistic semantics, there is not even one instance of a $\beta \wedge \neg\alpha$-outcome which is preferred to all $\alpha \wedge \neg\beta$-outcomes and (2) for pessimistic semantics, there is not even one instance of an $\alpha \wedge \neg\beta$-outcome which is less preferred to all $\neg\alpha \wedge \beta$-outcomes. Therefore strong, ceteris paribus, optimistic and pessimistic semantics satisfy P1. In the case of opportunistic semantics, because of its loose requirements, all we can say is that it is possible that $\beta \rhd \alpha$. Therefore opportunistic semantics does not satisfy P1, as illustrated by the following counterexample:

> Consider $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking}, \mathsf{Bottom\text{-}ranking}\}$ and the preference statement *London* $\rhd_{opp}$ *Paris*.
>
> $$\begin{array}{llll} \text{London + Top-ranking} & \approx & \text{Paris + Top-ranking} & \\ \quad \succ \ \text{London + Bottom-ranking} & \approx & \text{Paris + Bottom-ranking} & \end{array}$$
>
> satisfies *London* $\rhd_{opp}$ *Paris* but equally satisfies *Paris* $\rhd_{opp}$ *London*.

P2: A given semantics satisfies P2 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ then if $\alpha \equiv \alpha'$ then $\alpha' \rhd \beta$ and if $\beta \equiv \beta'$ then $\alpha \rhd \beta'$. With the exception of ceteris paribus semantics, we can say for all the other semantics (amongst the five) that if $\alpha \rhd \beta$ then we can ensure that α (resp. β) can be safely replaced by an equivalent formula α' (resp. β') in $\alpha \rhd \beta$. This is because any $\alpha \wedge \neg\beta$-outcome can be compared to every $\neg\alpha \wedge \beta$-outcome without depending on the variables that do not appear in either outcome. Therefore, an equivalent description of α or β will not change the result of this comparison. It ensues that strong, optimistic, pessimistic and opportunistic semantics satisfy P2. In the case of ceteris paribus semantics, the comparison between $\alpha \wedge \neg\beta$- and $\neg\alpha \wedge \beta$-outcomes depends on the variables that do not appear in either outcome. Thus replacing α (resp. β) by α' (resp. β') does not ensure keeping the same variables that do not appear in either outcome and could therefore change the result of this comparison. [2] As illustrated by the following counterexample, ceteris paribus semantics does not satisfy P2:

> Consider $\Omega = \{\mathrm{London}, \mathrm{Paris}\} \times \{\mathrm{Top\text{-}ranking}, \mathrm{Bottom\text{-}ranking}\}$.
> Given p_1 = *London* $\rhd_{cp}$ *Paris* and
> p_2 = *London* $\wedge$ *(London* $\vee$ *Top-ranking)* $\rhd_{cp}$ *Paris*
> (resp. p_3 = *London* $\rhd_{cp}$ *Paris* $\wedge$ *(Paris* $\vee$ *Top-ranking)*),
>
> London + Top-ranking
> $\succ$ Paris + Top-ranking
> $\succ$ London + Bottom-ranking
> $\succ$ Paris + Bottom-ranking
>
> satisfies p_1 but does not satisfy p_2 (resp. p_3) .

P3: A given semantics satisfies P3 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \gamma$ and $\succeq \models_S \beta \rhd \gamma$ then $\succeq \models_S \alpha \vee \beta \rhd \gamma$. In the case of strong semantics, we can affirm that any $\alpha \wedge \neg\gamma$ (resp. $\beta \wedge \neg\gamma$)-outcome is preferred to all $\neg\alpha \wedge \gamma$ (resp. $\neg\beta \wedge \gamma$)-outcomes. This means, in particular, that all $\alpha \wedge \neg\gamma$ (resp. $\beta \wedge \neg\gamma$)-outcomes are preferred to $\neg\alpha \wedge \neg\beta \wedge \gamma$-outcomes. This is nothing but all $(\alpha \wedge \neg\gamma) \vee (\beta \wedge \neg\gamma)$-outcomes preferred to $\neg\alpha \wedge \neg\beta \wedge \gamma$-outcomes, or $(\alpha \vee \beta) \wedge \neg\gamma$-outcomes preferred to $\neg\alpha \wedge \neg\beta \wedge \gamma$-outcomes. This is equivalent to the statement $(\alpha \vee \beta) \rhd \gamma$ and therefore strong semantics satisfies P3.

By Table 3.3 and Proposition 1, optimistic semantics is left-expansion and right-reduction tolerant and therefore satisfies P3.

To prove that ceteris paribus, pessimistic and opportunistic semantics do not satisfy P3, we supply the following counterexamples:

> $\Omega = \{\mathrm{London}, \mathrm{Paris}\} \times \{\mathrm{Top\text{-}ranking(Top)}, \mathrm{Bottom\text{-}ranking(Bot)}\} \times \{\mathrm{Science}, \mathrm{Arts}\}$, p_1 = *London* $\rhd$ *Arts*, p_2 =*Top* $\rhd$ *Arts*, and p_3 =*London* $\vee$ *Top* $\rhd$ *Arts*.

2. This is also the reason why this semantics does not satisfy any of the other postulates either, with the exception of P7.

1. The preference relation

$$\begin{aligned} &\text{London + Bot + Science} \approx \text{Paris + Top + Science} \approx \text{London + Top + Arts} \\ &\succ \text{Paris + Bot + Arts} \approx \text{London + Top + Science} \\ &\succ \text{Paris + Bot + Science} \approx \text{Paris + Top + Arts} \approx \text{London + Bot + Arts} \end{aligned}$$

   satisfies *London* $\rhd_{cp}$ *Arts* and *Top-ranking* $\rhd_{cp}$ *Arts* but does not satisfy *London* $\vee$ *Top-ranking* $\rhd_{cp}$ *Arts*. This proves that ceteris paribus semantics does not satisfy P3.

2. The preference relation

$$\begin{aligned} &\text{London + Top + Science} \approx \text{London + Top + Arts} \approx \text{Paris + Bot + Science} \\ &\approx \text{Paris + Bot + Arts} \approx \text{London + Bot + Science} \approx \text{Paris + Top + Science} \\ &\succ \text{Paris + Top + Arts} \approx \text{London + Bot + Arts} \end{aligned}$$

   satisfies *London* $\rhd_{pes}$ *Arts* and *Top-ranking* $\rhd_{pes}$ *Arts* but does not satisfy *London* $\vee$ *Top-ranking* $\rhd_{pes}$ *Arts*. This proves that pessimistic semantics does not satisfy P3. This also proves that opportunistic semantics does not satisfy P3.

P4: A given semantics satisfies P4 if for all $\succeq$ such that $\succeq \models_S \alpha \vee \beta \rhd \gamma$ then $\succeq$ satisfies $\alpha \rhd \gamma$ and $\beta \rhd \gamma$. By Table 3.3 and Proposition 1, pessimistic semantics is left-reduction and right-expansion tolerant and therefore satisfies P4.

To prove that strong, ceteris paribus, optimistic and opportunistic semantics do not satisfy P4, we supply the following counterexamples:

1. $\Omega = \{\text{London}, \text{Paris}\} \times \{\text{Top-ranking}(\text{Top}), \text{Bottom-ranking}(\text{Bot})\} \times \{\text{Science}, \text{Arts}\}$,
   p_1 = *London* $\vee$ *Top-ranking* $\rhd_{st}$ *Science*, p_2 = *London* $\rhd_{st}$ *Science*,
   p_3 = *Top-ranking* $\rhd_{st}$ *Science*. The preference relation

$$\begin{aligned} &\text{London + Top + Arts} \approx \text{London + Bot + Arts} \approx \text{Paris + Top + Arts} \\ &\approx \text{Paris + Top + Science} \approx \text{London + Bot + Science} \\ &\succ \text{Paris + Bot + Science} \approx \text{Paris + Bot + Arts} \approx \text{London + Top + Science} \end{aligned}$$

   satisfies p_1 but does not satisfy either p_2 or p_3. This proves that strong semantics does not satisfy P4. This also proves that ceteris paribus and optimistic semantics do not satisfy P4 either.

2. $\Omega = \{\text{London}, \text{Paris}\} \times \{\text{Top-ranking}(\text{Top}), \text{Bottom-ranking}(\text{Bot})\}$,
   p_1 = *(London* $\wedge$ *Top)* $\vee$ *(London* $\wedge$ *Bot)* $\rhd_{opp}$ *Paris* $\wedge$ *Top*,
   p_2 = *London* $\wedge$ *Bot* $\rhd_{opp}$ *Paris* $\wedge$ *Top*.

$$\begin{aligned} &\text{London + Top-ranking} \\ &\succ \text{Paris + Top-ranking} \quad \approx \quad \text{London + Bottom-ranking} \\ &\succ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that opportunistic semantics does not satisfy P4.

P5: A given semantics satisfies P5 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ and $\succeq \models_S \alpha \rhd \gamma$ then $\succeq$ satisfies $\alpha \rhd \beta \vee \gamma$. In the case of strong semantics, we can affirm that any $\alpha \wedge \neg\beta$ (resp. $\alpha \wedge \neg\gamma$)-outcome is preferred to all $\neg\alpha \wedge \beta$ (resp. $\neg\alpha \wedge \gamma$)-outcomes. This means, in particular, that all $\alpha \wedge \neg\beta \wedge \neg\gamma$-outomes are preferred to $\neg\alpha \wedge \beta$ (resp. $\neg\alpha \wedge \gamma$)-outcomes. This is nothing but all $\alpha \wedge \neg\beta \wedge \neg\gamma$-outcomes preferred to $(\neg\alpha \wedge \beta) \vee (\neg\alpha \wedge \gamma)$-outcomes, or all $\alpha \wedge \neg\beta \wedge \neg\gamma$-outcomes preferred to $\neg\alpha \wedge (\beta \vee \gamma)$. This is equivalent to the statement $\alpha \rhd (\beta \vee \gamma)$ and therefore strong semantics satisfies P5.

By Table 3.3 and Proposition 1, pessimistic semantics is left-reduction and right-expansion tolerant and therefore satisfies P5.

To prove that ceteris paribus, optimistic and opportunistic semantics do not satisfy P5, we supply the following counter-examples:

1. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking(Top)}, \mathsf{Bottom\text{-}ranking(Bot)}\} \times \{\mathsf{Science}, \mathsf{Arts}\}$, p_1 = *London* $\rhd_{cp}$ *Top-ranking*,
   p_2 = *London* $\rhd_{cp}$ *Science*,
   p_3 = *London* $\rhd_{cp}$ *Top-ranking* $\vee$ *Science*.

   $$\begin{aligned} &\text{London + Top + Science} \approx \text{London + Bot + Science} \approx \text{London + Top + Arts} \\ &\quad\succ \ \text{Paris + Top + Science} \\ &\quad\succ \ \text{London + Bot + Arts} \\ &\quad\succ \ \text{Paris + Bot + Science} \approx \text{Paris + Top + Arts} \approx \text{Paris + Bot + Arts} \end{aligned}$$

   satisfies p_1 and p_2 but does not satisfy p_3. This proves that ceteris paribus semantics does not satisfy P5.
2. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking(Top)}, \mathsf{Bottom\text{-}ranking(Bot)}\} \times \{\mathsf{Science}, \mathsf{Arts}\}$, p_1 = *London* $\rhd_{opt}$ *Top-ranking*, p_2 = *London* $\rhd_{opt}$ *Science* and
   p_3 = *London* $\rhd_{opt}$ *Top-ranking* $\vee$ *Science*.

   $$\begin{aligned} &\text{London + Top + Arts} \approx \text{London + Bot + Science} \\ &\quad\succ \ \text{Paris + Top + Science} \approx \text{Paris + Top + Arts} \approx \text{London + Bot + Arts} \\ &\quad\approx \ \text{Paris + Bot + Science} \approx \text{Paris + Bot + Arts} \approx \text{London + Top + Science} \end{aligned}$$

   satisfies p_1 and p_2 but does not satisfy p_3. This proves that optimistic semantics does not satisfy P5. This also proves that opportunistic semantics does not satisfy P5 either.

P6: A given semantics satisfies P6 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta \vee \gamma$ then $\succeq$ satisfies $\alpha \rhd \beta$ and $\alpha \rhd \gamma$. By Table 3.3 and Proposition 1, optimistic semantics is left-expansion and right-reduction tolerant and therefore satisfies P6.

To prove that strong, ceteris paribus, pessimistic and opportunistic semantics do not satisfy P6, we supply the following counter-examples:

1. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking(Top)}, \mathsf{Bottom\text{-}ranking(Bot)}\} \times \{\mathsf{Science}, \mathsf{Arts}\}$, p_1 = *London* $\rhd_{\mathtt{st}}$ *Top-ranking* $\vee$ *Science*, p_2 = *London* $\rhd_{\mathtt{st}}$ *Science* and p_3 = *London* $\rhd_{\mathtt{st}}$ *Top-ranking*.

$$\begin{aligned} &\text{London + Bot + Arts} \\ &\quad \succ \text{ Paris + Top + Science} \approx \text{Paris + Top + Arts} \approx \text{Paris + Bot + Science} \\ &\quad \approx \text{ London + Bot + Science} \approx \text{London + Top + Arts} \approx \text{Paris + Bot + Arts} \\ &\quad \approx \text{ London + Top + Science} \end{aligned}$$

   satisfies p_1 but does not satisfy either p_2 or p_3. This proves that strong semantics does not satisfy P6. This also proves that ceteris paribus and pessimistic semantics do not satisfy P6 either.
2. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking}), \mathsf{Bottom\text{-}ranking})\}$, p_1 = *London* $\wedge$ *Top* $\rhd_{\mathtt{opp}}$ *(Paris* $\wedge$ *Top-ranking)* $\vee$ *(Paris* $\wedge$ *Bottom-ranking)*, p_2 = *London* $\wedge$ *Top* $\rhd_{\mathtt{opp}}$ *Paris* $\wedge$ *Bottom*.

$$\begin{aligned} \text{London + Top-ranking} &\approx \text{Paris + Bottom-ranking} \\ \succ \text{ Paris + Top-ranking} &\approx \text{London + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that opportunistic semantics does not satisfy P6.

P7: A given semantics satisfies P7 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ then $\succeq$ satisfies $\alpha \vee \gamma \rhd \beta \vee \gamma$. Strong and ceteris paribus semantics are left- and right-reduction tolerant and therefore satisfy P7.

To prove that optimistic, pessimistic and opportunistic semantics do not satisfy P7, we supply the following counterexamples:

1. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking(Top)}, \mathsf{Bottom\text{-}ranking(Bot)}\}$, p_1 = *London* $\wedge$ *Top* $\rhd_{\mathtt{opt}}$ *Paris* $\wedge$ *Bot*, p_2 = *(London* $\wedge$ *Top)* $\vee$ *(Paris* $\wedge$ *Top)* $\rhd_{\mathtt{opt}}$ *(Paris* $\wedge$ *Bot)* $\vee$ *(Paris* $\wedge$ *Top)*.

$$\begin{aligned} \text{London + Top-ranking} &\approx \text{Paris + Top-ranking} \\ \succ \text{ London + Bottom-ranking} &\approx \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that optimistic semantics does not satisfy P7.
2. $\Omega = \{\mathsf{London}, \mathsf{Paris}\} \times \{\mathsf{Top\text{-}ranking(Top)}, \mathsf{Bottom\text{-}ranking(Bot)}\}$, p_1 = *Paris* $\wedge$ *Top* $\rhd_{\mathtt{pes}}$ *London* $\wedge$ *Bot*, p_2 = *(Paris* $\wedge$ *Top)* $\vee$ *(Paris* $\wedge$ *Bot)* $\rhd_{\mathtt{pes}}$ *(London* $\wedge$ *Bot)* $\vee$ *(Paris* $\wedge$ *Bot)*.

$$\begin{aligned} &\text{London + Top-ranking} \approx \text{Paris + Top-ranking} \\ &\quad \succ \text{ London + Bottom-ranking} \\ &\quad \succ \text{ Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that pessimistic semantics does not satisfy P7.

3. $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}(\mathtt{Top}), \mathtt{Bottom\text{-}ranking}(\mathtt{Bot})\} \times \{\mathtt{Science}, \mathtt{Arts}\}$,
   p_1 = *London* $\triangleright_{opp}$ *Top-ranking*,
   p_2 = *London* $\vee$ *Science* $\triangleright_{opp}$ *Top-ranking* $\vee$ *Science*.

$$\begin{aligned} &\text{London + Top + Arts} \approx \text{London + Bot + Arts} \\ &\quad\approx \text{London + Bot + Science} \approx \text{Paris + Top + Arts} \\ &\succ \text{Paris + Top + Science} \approx \text{Paris + Bot + Science} \\ &\quad\approx \text{London + Top + Science} \approx \text{Paris + Bot + Arts} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that opportunistic semantics does not satisfy P7.

P8: A given semantics satisfies P8 if for all $\succeq$ such that $\succeq \models_S \alpha \triangleright \beta$ and $\mathrm{Mod}(\alpha') \subset \mathrm{Mod}(\alpha)$ then $\succeq$ satisfies $\alpha' \triangleright \beta$.

By Table 3.3 and Proposition 1, pessimistic semantics is left-reduction and right-expansion tolerant and therefore satisfies P8.

To prove that strong, ceteris paribus, optimistic and opportunistic semantics do not satisfy P8, we supply the following counter-examples:

1. $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}(\mathtt{Top}), \mathtt{Bottom\text{-}ranking}(\mathtt{Bot})\} \times \{\mathtt{Science}, \mathtt{Arts}\}$,
   p_1 = *London* $\vee$ *Top-ranking* $\triangleright_{st}$ *Science*,
   p_2 = *London* $\triangleright_{st}$ *Science*.

$$\begin{aligned} &\text{London + Top + Arts} \approx \text{London + Bot + Arts} \approx \text{Paris + Top + Arts} \\ &\quad\approx \text{London + Bot + Science} \approx \text{Paris + Top + Science} \\ &\succ \text{Paris + Bot + Science} \approx \text{London + Top + Science} \approx \text{Paris + Bot + Arts} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that strong semantics does not satisfy P8.

2. $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}, \mathtt{Bottom\text{-}ranking}\}$,
   p_1 = *London* $\triangleright_{cp}$ *Paris*,
   p_2 = *London* $\wedge$ *Bottom-ranking* $\triangleright_{cp}$ *Paris*.

$$\begin{aligned} &\text{London + Top-ranking} \\ &\succ \text{Paris + Top-ranking} \approx \text{London + Bottom-ranking} \\ &\succ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that ceteris paribus semantics does not satisfy P8. This also proves that optimistic semantics do not satisfy P8 either.

3. $\Omega = \{\mathtt{London}, \mathtt{Paris}\} \times \{\mathtt{Top\text{-}ranking}, \mathtt{Bottom\text{-}ranking}\}$,
   p_1 = *London* $\triangleright_{opp}$ *Paris*,

p_2 = *London* $\wedge$ *Bottom-ranking* $\rhd_{\text{opp}}$ *Paris*.

$$\begin{aligned} &\text{London + Top-ranking} \\ &\quad \succ \text{Paris + Top-ranking} \approx \text{London + Bottom-ranking} \\ &\quad\quad \approx \text{Paris + Bottom-ranking} \end{aligned}$$

satisfies p_1 but does not satisfy p_2. This proves that opportunistic semantics does not satisfy P8.

P9: A given semantics satisfies P9 if for all $\succeq$ such that $\succeq \models_S \alpha \rhd \beta$ and $\mathrm{Mod}(\beta') \subset \mathrm{Mod}(\beta)$ then $\succeq$ satisfies $\alpha \rhd \beta'$.

By Table 3.3 and Proposition 1, optimistic semantics is left-expansion and right-reduction tolerant and therefore satisfies P9.

To prove that strong, ceteris paribus, pessimistic and opportunistic semantics do not satisfy P9, we supply the following counter-examples:

1. $\Omega = \{\mathrm{London}, \mathrm{Paris}\} \times \{\mathrm{Top\text{-}ranking(Top)}, \mathrm{Bottom\text{-}ranking(Bot)}\} \times \{\mathrm{Science}, \mathrm{Arts}\}$,
   p_1 = *London* $\rhd_{\text{st}}$ *Top-ranking* $\vee$ *Science* ,
   p_2 = *London* $\rhd_{\text{st}}$ *Top-ranking*.

   $$\begin{aligned} &\text{London + Top + Arts} \approx \text{London + Bot + Arts} \\ &\quad \succ \text{London + Bot + Science} \approx \text{Paris + Top + Arts} \approx \text{Paris + Top + Science} \\ &\quad \succ \text{Paris + Bot + Science} \approx \text{London + Top + Science} \approx \text{Paris + Bot + Arts} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that strong semantics does not satisfy P9. This also proves that pessimistic semantics do not satisfy P9 either.
2. $\Omega = \{\mathrm{London}, \mathrm{Paris}\} \times \{\mathrm{Top\text{-}ranking}, \mathrm{Bottom\text{-}ranking}\}$,
   p_1 = *London* $\rhd_{\text{cp}}$ *Paris*,
   p_2 = *London* $\rhd_{\text{cp}}$ *Paris* $\wedge$ *Top-ranking* .

   $$\begin{aligned} &\text{London + Top-ranking} \\ &\quad \succ \text{Paris + Top-ranking} \approx \text{London + Bottom-ranking} \\ &\quad \succ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that ceteris paribus semantics does not satisfy P9.
3. $\Omega = \{\mathrm{London}, \mathrm{Paris}\} \times \{\mathrm{Top\text{-}ranking}, \mathrm{Bottom\text{-}ranking}\}$,
   p_1 = *London* $\rhd_{\text{cp}}$ *Paris*,
   p_2 = *London* $\rhd_{\text{cp}}$ *Paris* $\wedge$ *Top-ranking* .

   $$\begin{aligned} &\text{London + Top-ranking} \approx \text{Paris + Top-ranking} \\ &\quad \approx \text{London + Bottom-ranking} \\ &\quad \succ \text{Paris + Bottom-ranking} \end{aligned}$$

   satisfies p_1 but does not satisfy p_2. This proves that opportunistic semantics does not satisfy P9.

□

This completes our postulate-based analysis of comparative preference statements. We conclude by summing up.

Conclusion

This chapter presents a thorough investigation of comparative preference statements. Beginning with a rigorous exposition of their formulation and reasoning mechanisms, it concludes with our own contribution: a postulate-based analysis which helps understand the behaviour of the different semantics.

The results of our study reveal in particular that opportunistic semantics has bad properties as concerns both, the reasoning mechanisms to compute distinguished preference relations and the postulate-based analysis. This is not surprising as it is the weakest semantics. That said, it is useful in other frameworks such as interval orders (The and Tsoukiàs, 2005) and therefore calls for a further investigation of its properties.

From Table 3.4, we know that strong semantics is coherent, syntax independent and it ensures that (i) $\alpha \vee \beta \rhd \gamma$ entails $\alpha \rhd \gamma$ and $\beta \rhd \gamma$, (ii) $\alpha \rhd \beta$ and $\alpha \rhd \gamma$ entail $\alpha \rhd \beta \vee \gamma$ and (iii) $\alpha \rhd \beta$ entails $\alpha \vee \gamma \rhd \beta \vee \gamma$. Ceteris paribus does not satisfy many postulates. It only ensures coherence and preference independence and thus does not allow any decomposition/composition. Lastly, we said earlier (§3.2.1) that optimistic and pessimistic semantics exhibit a dual behaviour. This property is reflected in Table 3.4. While optimistic semantics allows left composition, right decomposition and right weakening, pessimistic semantics allows left decomposition, right composition and left weakening. This duality corroborates existing works on bipolar information (Benferhat et al., 2002b).

In the next chapter, we shall use the content of this chapter as a primary motivation to design a personalised recommender system using comparative preference statements. We shall also exploit the bipolar aspect of optimistic and pessimistic semantics. To elicit preferences following these different semantics, the results of Chapter 2 will guide the design of an interactive preference elicitation unit which will be included in the recommender system.

4

A Framework for Personalised Decision Support using Comparative Preference Statements

Introduction

We now come to the unifying chapter of this thesis, where the results of all the previous studies are applied to the design of a framework for personalised recommendation using comparative preference statements. We introduce our approach by first outlining our arguments, based on the literature review presented in Chapter 1.

Motivations

Why Recommend? Information search and selection is increasingly performed online, and recommender systems have proven to be a valuable way for online users to cope with information overload. They have become popular tools in electronic commerce, and are found in practically every virtual interaction we have. They use algorithms that approximate, or predict, possible recommendations on *items*, based on available information about the users and/or the items in question. They generate *recommendation sets* on the basis of *similarity measures* to compute similar users and/or similar items or *predicted ratings* on items. The growing number of users, content and social media have provided a fertile ground for the improvement of current approaches, and the state of the art today performs remarkable feats of artificial intelligence.

Among traditional approaches, user- and content-based approaches are *quantitative*, as they depend on and manipulate data that is quantified as weights (for user preferences, ratings), or similarity measures (for users and content). Knowledge-based approaches can use *qualitative* information obtained from users to guide them through the decision process. The former are more efficient but only provide predictions, while the latter adhere closely to the decision problem and provide exact solutions, but at the cost of complex algorithms.

Why Preference-based Recommendation? The study of preferences in AI has explored different aspects of handling preferences: *acquisition, modelling, compact representation, reasoning*. Different approaches in each category have been proposed and analysed, with arguments for the specific purposes they are most suited to.

The body of work reflects the numerous possibilities of applying these for decision support systems such as preference-based web applications e.g. product search, recommender systems, personal assistant agents, and personalised user interfaces, stressing upon the growing importance of user-involved preference acquisition and recommendation.

The inclusion of techniques in preference learning, preference elicitation and preference based ranking into recommender systems shows the relevance of preference-based recommendation. Typical examples showing good results are books, movies, music, news, etc., or items for which the assumption that user preferences do not change dramatically is valid.

On the other hand, decision-making tasks where user preferences can be complex—which would require careful consideration of available options such as buying a house, a car, etc.; or where making a choice can depend on several conditions which may vary every time, such as travel plans, hotel booking, etc.—could be addressed through the use of preferences in knowledge-based recommender systems; for which there still remain hitherto untapped directions which could enrich the state-of-the-art.

Specifically, theoretical research in AI has shown ways of handling preferences about items to induce an ordering on the entire set of items, thereby generating an exact recommendation on them, as opposed to predicting an approximate recommendation on them based on similarity measures or predicted ranks. Among the traditional approaches in recommender systems, these could be used to improve the state-of-the-art in knowledge-based recommender systems.

Why Comparative Preference Statements? The representation of user preferences as comparative preference statements is intuitively similar to the way users express them in natural language, as we saw in our linguistic study (Kaci et al., 2014).

CP-nets are a popular form of such representation: a graphical formalism for representing qualitative conditional preference statements and their interdependencies. They have successfully been implemented in recommender systems using preference learning approaches.

The inclusion of comparative preference theories in information recommendation by the means of conversational recommender systems (Trabelsi et al., 2011) shows the relevance of exploiting this expressive approach of preference representation.

We follow this research direction, and consider including those semantics associated with comparative preference statements that have not yet been implemented in recommender systems. The additional semantic diversity offered by this form of representing user preferences seems to be a promising outlook for personalised recommendation, given the positive results of Trabelsi et al. (2011). Moreover, existing theoretical research on the topic reveals the possibility of computing solutions to satisfy user preferences, even in the presence of complex preferences such as defeasible preferences, or preferences that are inconsistent when taken together. Deploying such methods for personalised recommendation presents an equally interesting avenue for further research in the topic.

Our Approach

Taking up the afore-seen arguments for (1) preference elicitation, (2) using comparative preference statements, and (3) expanding the state-of-the-art in knowledge-based recommender systems, we develop our approach focussing on the logical representation of comparative preference statements with associated semantics.

We orient the design of our framework towards that of recommender systems, using a hybrid approach which borrows elements from each of the traditional approaches:

— user-based (preferences of similar users),
— content-based (preferences about similar features), and
— knowledge-based (to reason about items which respond to user requirements).

Our primary approach is that of knowledge-based systems, since they can use *qualitative* information obtained from users to guide them through the decision process. They generally use a reasoning mechanism, or a knowledge engineering AI, to generate the resulting recommendation set. In our system, this reasoning mechanism is a *logic-based AI* which manipulates preferences to compute an ordered set of items. The order is induced by the preferences, and the recommendation set is therefore the preferred items in the ordered set.

The extensive study of comparative preference statements provided in Chapter 3 is the theoretical basis for this recommendation AI which computes recommendations and verifies that the generated solutions are satisfactory. The formal language used for this engine is based on that of modelling preferences and formulating comparative preference statements; the algorithms for computing recommendations are an exact implementation of those described.

We tailor our system to this reasoning engine with the help of machine-learning techniques for data analysis, as found in user- and content-based approaches. This combines the efficiency of statistically-driven AI with the accuracy of logic-based AI in achieving our results. We therefore have a specifically designed preprocessing unit for *item-profiling*.

Finally, we favour the active elicitation of preferences from users, as in explicit elicitation-based approaches, and include a *graphical web-interface* by which users can be guided to express their preferences. Our design borrows concepts from (1) our previous linguistic study (Chapter 2) and (2) bipolar preferences so that elicited preferences can be associated with the different semantics from AI research.

We adapt the elicitation protocol presented at the end of our linguistic study, bearing in mind that interaction should be reduced to a minimum, without losing out on the expressivity of preferences. We therefore avoid asking the user to express their preferences in NL, and design an alternative graphical interaction which adheres to the expressivity of NL preferences. Specifically, we consider the set of linguistic markers and identifiers from Table 2.9 on page 47 from our linguistic study and find a corresponding graphical user interaction to express the same preference. For semantics that were found to be implicitly expressed in NL, we use cognitive notions from bipolar preferences to distinguish these semantics.

Overall, the architecture of our proposed system can be summarised as: (1) a preprocessing unit for *item-profiling*, or detecting positive and negative features from existing user reviews about items; (2) an interactive unit for *preference elicitation*, or acquiring the user's current preferences about these features; (3) *computation of recommendations* on the item profiles based on the elicited information using a preference logic-based reasoning engine; and (4) the final recommendation unit which resolves instances of empty/too large recommendation sets.

Outline of the Chapter We begin by defining the formal language upon which rests the AI used in our recommendation engine (§4.1). We then describe our system architecture through its implementation in a specific scenario (§4.2). We conclude by presenting the results of our implementation with a discussion (§4.3) of the proposed framework for personalised recommendation.

4.1 Background

Among the various approaches for predicting recommendations from a set of items, knowledge-based approaches use a reasoning mechanism, or a knowledge engineering AI, to generate the resulting set.

In our system, this reasoning mechanism is a logic-based AI which manipulates preferences to compute an ordered set of items. The order is induced by the preferences, and the recommendation set is therefore the preferred items in the ordered set.

We begin this section by formalising a *preference language* to describe both preferences and items, and *preference relations* to characterise the recommendation set as an ordered set of items. We then discuss the necessary steps involved in using the former to compute the latter.

4.1.1 The Preference Language

To ensure a linguistic coherence between describing preferences and the items that satisfy these preferences, we use the same language to describe both preferences and items. By formalising the notion of *item features* as the common factor between preferences and items, we can describe (1) items in terms of their features, and (2) preferences on these features.

To define the syntax of this preference language, we consider the general scheme of compactly describing a preference model as described in § 1.2.1 on page 15 and adapt it to our framework using recommendation terminology. We thus formally represent items and preferences in our preference language $\mathcal{L}$ as following:

— A finite set Σ of *features* which describe characteristics of an item,
— A finite set called the *domain*, $\mathrm{Dom}(X)$, for each feature X in Σ. For the sake of simplicity, and without loss of generality, $\mathrm{Dom}(X)$ is the boolean set $\{\mathtt{true},\ \mathtt{false}\}$[1].
— An *item* ω, which is the result of assigning a value to each feature X in Σ,
— The set of all *items* Ω (i.e. the cartesian product of all features in Σ). In practice, this is the set of all *feasible* items (we assume that integrity constraints can be defined for items that are not feasible).
— *Logical Formulae* are built by combining logical propositions using the standard logical connectors $\wedge$, $\vee$ and $\neg$. By abuse of language, we shall say that for each feature $X \in \Sigma$, we have the corresponding logical proposition X.

1. A feature having an n-ary domain can always be redefined as n features with binary domains: Given $\mathrm{Dom}(X) = \{x_1, ..., x_n\}$, X can be equivalently replaced by n new features $X_1, ...X_n$ where $X_i = x_i$ and $\mathrm{Dom}(X_i) = \{\mathtt{true},\ \mathtt{false}\}$.

— *Preference Statements* express a preference between two logical formulae.

We say that a given preference statement "p preferred to q" can be interpreted as a preference over items "ω preferred to ω'" when ω *satisfies* p and ω' *satisfies* q, as defined in Definition 6 on page 16.

4.1.2 Preference Relations

Our system rests upon the ability to compute an order on the set of items, based on how these items satisfy, or not, a set of preference statements. We therefore look at the set of items Ω, and formalise the notions of ordering it according to preferences.

A preference relation is defined on the set Ω, and evaluates the pairwise comparison of items $\omega, \omega' \in \Omega$ following Definition 3 on page 10.

A given preference relation being reflexive and transitive, we say that it induces a *partial preorder* on Ω. It induces a *total preorder* on Ω when there are no incomparable items, and Ω can be partitioned into equivalence classes E_i satisfying:

1. $\forall i = 1...n, \quad E_i \neq \emptyset$,
2. $E_1 \uplus ... \uplus E_n = \Omega$,
3. $\forall \omega, \omega' \in E_i, \quad \omega \approx \omega'$.

We then have an ordered partition of Ω given $\succeq$, written as $(E_1, ..., E_n)$, if and only if the following condition holds:

$$\forall \omega, \omega' \in \Omega, \quad \omega \succ \omega' \iff \omega \in E_i,\ \omega' \in E_j \text{ with } i < j.$$

This is the structure we seek to construct through our reasoning engine.

We now look at how we intend to use preference statements expressed in $\mathcal{L}$ to define preference relations and consequently lead to computing recommendations.

4.1.3 From Preference Statements to Preference Relations

Our extensive study of comparative preference statements in the previous chapter (Ch.3) showed us that they are a compact, expressive and efficient way to describe preferences and compute preference relations induced by sets of these preferences. Additionally, our results in Chapter 2 showed us that it is possible to acquire user preferences expressed in natural language (NL) and formally represent them as comparative preference statements following strong, ceteris paribus, optimistic and pessimistic semantics.

We therefore select this form of representation to formally describe preferences in our preference language $\mathcal{L}$. We use Definition 10 (comparative preference statement) to define our preference statements, and associate these with four preference semantics from Definition 11. We now repeat a few pertinent definitions here for ease of reference.

Definition (Preference Semantics [Def.11]). Let $p \triangleright q$ be a comparative preference statement and $\succeq$ a preference relation. We say that $p \triangleright q$ is associated with S-semantics for $S \in \{st, cp, opt, pes\}$ (or $p \triangleright q$ is an *S-preference*), and we write $p \triangleright_S q$ when:

— for $S = st : \forall\omega \in Mod(p \wedge \neg q), \forall\omega' \in Mod(\neg p \wedge q),\ \omega \succ \omega'$;

— for $S = cp : \forall\omega \in Mod(p \wedge \neg q), \forall\omega' \in Mod(\neg p \wedge q),\ \omega \succ \omega'$ if the two items have the same valuation over features that change the truth-values of p and q;

— for $S = opt : \exists\omega \in Mod(p \wedge \neg q), \forall\omega' \in Mod(\neg p \wedge q),\ \omega \succ \omega'$;

— for $S = pes : \exists\omega' \in Mod(\neg p \wedge q), \forall\omega \in Mod(p \wedge \neg q),\ \omega \succ \omega'$.

We then say that $\succeq$ satisfies $p \triangleright_S q$.

Definition (Preference Set [Def.12]). A set of S-preferences for $S \in \{st, cp, opt, pes\}$, is defined as $\mathcal{P}_S = \{\alpha \triangleright_S \beta\}$. A *preference set*, in general, is denoted by $\mathcal{P}_\triangleright$ when it contains preferences associated with several semantics. Thus, $\mathcal{P}_\triangleright = \mathcal{P}_{st} \uplus \mathcal{P}_{cp} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes} \uplus \mathcal{P}_{opp}$.

To formalise the correspondence between preference statements and preference relations, we say that:

Definition (Preference Set Consistency [Def.13]). A preference relation is a *model* of $\mathcal{P}_\triangleright$ if and only if it satisfies each comparative preference statement $p \triangleright_S q$ in $\mathcal{P}_\triangleright$. A preference set is *consistent* if and only if it has a model.

In the context of decision-support, generating a recommendation translates to computing a preference relation $\succeq$, or an ordering of the set of items, which is a model of $\mathcal{P}_\triangleright$.

At this point, let us recall our discussion in Chapter 3 (§ 3.2.1 on page 66) about using the non-monotonic principle of specificity (Def.14) to resolve situations where preferences are uncertain, or must be revised based on subsequent information (e.g. defeasible preferences). We saw that in these situations, we must compute the *distinguished* models of the preference set, and described algorithms that do so.

We use these very algorithms in our reasoning engine to compute the ordered partitions corresponding to distinguished models of a given preference set $\mathcal{P}_\triangleright$. Our set of recommended items, then, is represented by the preferred items in these distinguished models.

Before we can formally define this set for our system, we must look at preferences from the user's point of view and discuss how we propose to integrate them into $\mathcal{P}_\triangleright$.

4.1.4 Types of User Preferences and their Formal Representation

Analysing preferences from the user's and the linguistic point of view, we have shown in Kaci et al. (2014) that user preferences expressed in natural language (NL) can be formally represented as comparative preference statements by defining a preference elicitation protocol to that effect.

We found that for a given user in a given decision scenario, NL-preference expressions often reflect preferences associated with different semantics, although not all were described explicitly: strong semantics was described explicitly, but the rest were implicitly implied). This showed the importance of having an elicitation protocol, as it was designed to extract even the implicit preferences from the user through additional interactions.

These implicit preferences were revealed in particular as expressions of (1) *conditional preferences* which are preferences expressed in a given context, and (2) *defeasible preferences*, which are preferences that get reversed in a given context. Taking examples from our study, "If it's for work, I'd like a laptop with a big screen" is a conditional preference and the two preferences in "I prefer small laptops in general, but if it's a Mac, I'd prefer a large one" are defeasible preferences.

Thus, when computing recommendations, we must account for the possibility that the set of user preferences, $\mathcal{P}_{\triangleright}$, may contain (1) conditional/defeasible preferences, and (3) the simultaneous use of different semantics.

Handling Conditional and Defeasible Preferences We first describe how conditional preferences can be expressed as comparative preference statements by manipulating the formulae in the preference. Conditional preference statements are expressed in $\mathcal{L}$ as preference statements of the form "When r, prefer p to q", where p, q, r are logical formulae. With the use of the logical connector $\wedge$, these can be equivalently described as comparative preference statements in the following way: "prefer $r \wedge p$ to $r \wedge q$". In this way, conditional preference statements can be captured using comparative preference statements in $\mathcal{P}_{\triangleright}$, without loss of semantic diversity.

Next, we address the handling of defeasible preferences. Since these are preferences which are reversed in a given context, they can be seen as two comparative preference statements: a general comparative preference statement $p \triangleright q$ and a more specific conditional preference statement $r \wedge q \triangleright r \wedge p$. Strong and ceteris paribus semantics are not suitable for handling defeasible preferences, while optimistic and pessimistic are (for details, see Kaci (2011, p.48-50)). Thus the elicitation of the latter two will allow us to handle defeasible preferences consistently.

Let us consider our example of choosing a university and the following defeasible preferences: The general preference "I prefer Paris to London" and the specific one "If the university offers an optional drama course, then I prefer London to Paris". In this situation, the two preferences can be consistently handled by processing the more specific one before the general one. The corresponding comparative preference statements read:

$$\mathit{Paris} \triangleright \mathit{London} \quad \text{and} \quad \mathit{drama} \wedge \mathit{London} \triangleright \mathit{drama} \wedge \mathit{Paris}.$$

Now, if we associate strong and ceteris paribus semantics to these statements, they result in contradictions since the first preference indicates that:

> All Paris universities must be preferred to all London universities, *including* those that are completed by an optional drama course.

and the second specifies that:

> All London universities completed by an optional a drama course must be preferred to all Paris universities completed by an optional drama course.

This shows that these two semantics are not suitable for handling defeasible preferences.

Considering the definitions of optimistic and pessimistic semantics (Def. 11), an ordering which satisfies both defeasible preferences consistently *can* respect both semantics. Continuing with our example, the following ordering satisfies both optimistic and pessimistic semantics and violates both strong and ceteris paribus semantics:

$$\begin{aligned} &\text{Paris + any course other than drama} \\ &\quad \succ \ \text{London + drama} \\ &\quad \succ \ \text{Paris + drama} \\ &\quad \succ \ \text{London + any course other than drama} \end{aligned}$$

Thus the elicitation of optimistic and pessimistic preferences will allow us to handle defeasible preferences consistently.

Handling the Simultaneous Use of Different Semantics Since optimistic and pessimistic preferences are suitable for handling defeasible preferences *and* capture strong and ceteris paribus preferences, it would actually suffice to design our system around optimistic and pessimistic preferences alone. However, the results of our previous linguistic study (see Table 2.11 on page 54) show that the most commonly expressed preferences in NL can be associated with strong or ceteris paribus semantics. The former semantics was mentioned explicitly, while the latter was identified to be implied implicitly. We therefore must account for this expressive diversity.

Considering the following result from Kaci and van der Torre (2008) (discussed in 3.2.1), it is possible to handle these semantics together:

> *According to specificity (Def. 14), there exists a unique total preorder which is the least specific model for strong, ceteris paribus and optimistic preferences taken together, and another which is the most specific model for strong, ceteris paribus and pessimistic preferences taken together.*

Thus, if we equip our system with the ability to elicit all the different semantics, we would be able to handle the preferences simultaneously in our reasoning engine by computing these two distinguished models.

In theory, this perfectly resolves the problem. In practice, eliciting semantics that are implicitly implied in preference expressions (i.e. ceteris paribus, optimistic and pessimistic) comes at the cost of further interactions with the user. To keep these interactions to a minimum without loss of expressive diversity, we chose to design the system around strong, optimistic and pessimistic semantics. In this way, we avoid the extra interaction for eliciting ceteris paribus semantics but maintain the expressive diversity since optimistic and pessimistic preferences capture this semantics anyway.

At this point, recalling from the conclusion of our in-depth study on comparative preference statements (§ 3.3.2 on page 93) that ceteris paribus semantics does not allow the decomposition/composition of preference statements (while strong semantics allows composition), our choice in designing the system around strong, optimistic and pessimistic semantics will allow for more flexibility in manipulating preference statements.

In our context, we define the two distinguished models that we must compute as the following:

Definition 17 (Strong-Optimistic and Strong-Pessimistic Recommendation Partitions)**.** Let $\mathcal{P}_{\rhd} = \mathcal{P}_{st} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes}$ be a preference set and Ω a set of items. Then the least- (resp. most-) specific model of the strong and optimistic (resp. pessimistic) preferences in $\mathcal{P}_{\rhd}$, $\succeq_{opt}$ (resp. $\succeq_{pes}$), is defined as a ***strong-optimistic (resp. strong-pessimistic) recommendation partition***, which is written as $(E_1, ..., E_n)$ (resp. $(E'_1, ..., E'_{n'})$).

Since our aim is to provide *one* recommendation set which corresponds to *all* preferences in $\mathcal{P}_{\rhd} = \mathcal{P}_{st} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes}$, these two models must be combined. This can be performed using a merging principle, which has been studied in particular for optimistic and pessimistic preferences in Kaci and van der Torre (2006). The final recommended set can therefore be computed by following any one of the merging principles described in this paper.

The method adopted for our system is a dictator merger called ***minimax merger*** in which dictatorship imposes that worlds are ordered by the 'dictates' of one relation (i.e. recommendation partition), and further refined, only in the case of equality, by the other one. The term minimax, as opposed to maximin, indicates which of the relations is given first priority. Thus $\succeq_{\mathtt{opt}}$ following 'min'imal specificity is favoured to $\succeq_{\mathtt{pes}}$ following 'max'imal specificity. We define the principle here, and shall discuss why we chose this particular one in §4.2.3.2.

Definition 18 (Minimax Merger)**.** Let $\succeq_1$ and $\succeq_2$ be two preorders on a set of items Ω. We define $\succeq$ as the ***minimax merger*** of $\succeq_1$ and $\succeq_2$ iff the following condition holds:

$$\forall \omega, \omega' \in \Omega, \omega \succ \omega' \text{ iff } \omega \succ_1 \omega' \text{ or } (\omega \approx_1 \omega' \text{ and } \omega \succ_2 \omega')$$

Considering the maximal (also known as the pareto-optimal) items in the final merged recommendation partition leads to defining the *recommendation set* proposed by the system to the user:

Definition 19 (Recommendation Set)**.** Let $\succeq$ be the minimax merger of a set of items Ω. Then the *recommendation set* Φ to be proposed to the user is defined as $\Phi = \{\omega \mid \omega \in \Omega, \nexists \omega' \in \Omega, \omega' \succ \omega\}$.

With this groundwork on preference handling, we can now describe a preference logic based approach to recommendation where user preferences are elicited and represented as $\mathcal{P}_{\triangleright}$ to compute an ordered partition and generate recommendations.

4.2 A Framework for Personalised Recommendation

In designing a recommender system which uses a recommendation engine based on the theory described in section 4.1, we define a system which contains the entire pipeline from data preprocessing to preference acquisition and manipulation to final recommendation. This is because (1) each recommendation scenario entails a specific database, and thereby a specific system adaptation, (2) users must be carefully guided to express their preferences so that they form a preference set which could be directly fed into the recommendation engine, (3) the recommendation engine must not only compute recommendations, it must also be able to resolve situations such as a large, or an empty recommendation set.

Borrowing elements from different traditional recommendation approaches, our first step is (1) *item-profiling*, as in content-based approaches. Our next step is (2) *preference elicitation*, as in explicit elicitation-based approaches. Our final step is (3) *computation of recommendations* on the item profiles based on elicited user requirements using a reasoning engine, as in knowledge-based approaches. Each of these steps is performed in a separate unit: (1) the data-preprocessing unit, (2) the preference elicitation unit, and (3) the recommendation engine.

4.2.1 Design Intuitions

The Preprocessing Unit. Our first intuition behind the design of the preprocessing phase of the system is to generate a database of items which can be described using the preference language $\mathcal{L}$ (see §4.1.1). This means constructing a database of items characterised by their features, each feature having a boolean domain.

Since recommendation datasets for different content such as news, movies, music, products, etc. vary considerably, we first define the content and adapt the entire system, i.e. instantiate the preference language $\mathcal{L}$, accordingly. Specifically, the sets Σ of content features and Dom(X) for each feature X are determined, so that each item $\omega \in \Omega$ can be described by assigning each of its features a boolean value. We call this step creating item profiles, or *item profiling*.

We distinguish between objective and subjective features: those which describe factual aspects such as technical specifications, and those about which users express subjective opinions in reviews. Thus, at an item-specific level, the boolean value for the former can be determined by its presence or absence in the item, while that for the latter can be determined in a two-step process: computing (1) *feature-level polarity* through positive and negative sentiment analysis of user expressions and (2) *item-specific polarity* by aggregating feature-level polarities for all occurrences of a given feature about a given item. We use several language engineering techniques to achieve this purpose.

Our second intuition for preprocessing the data is based on the nature of preferences and their inter-dependencies. Specifically, if inter-dependencies between features can be determined at a preprocessing phase, then these could be exploited for the formulation of conditional preferences and later on, to resolve situations such as a large recommendation set, or an empty one. We explore the data mining technique of association rule learning to discover dependencies between features.

The Preference Elicitation Unit. The challenge in designing our preference elicitation protocol lies in the dichotomy between (1) keeping interactions simple, intuitive and minimal and (2) ensuring the expressive diversity of the elicited information. To reconcile these two aspects, our intuition is to interact with the user through a simple and accessible *façade*, reserving all the theoretical steps to an *interpretation engine* that runs under the hood. This engine also makes use of the information retrieved during preprocessing to automate the formulation of preference statements when possible, and to optimise interactions when not.

Concerning the elicitation of comparative preference statements and their associated semantics, our idea is to guide the user to express their preference about each individual feature: each preference about a given feature X is interpreted as a comparative preference statement of the form $X \rhd_S \neg X$ (or $\neg X \rhd_S X$)

with $S \in \{\mathtt{st}, \mathtt{opt}, \mathtt{pes}\}$. We develop two graphical interactions to elicit strong, optimistic and pessimistic semantics, based on our linguistic and theoretical studies of these semantics.

The Recommendation Engine. Once the elicitation of a preference set $\mathcal{P}_{\triangleright}$ is complete, the intuition behind the reasoning engine is quite straightforward: we implement the algorithms that compute the preference relations $\succeq_{\mathtt{opt}}$ and $\succeq_{\mathtt{pes}}$ (described in §4.1) induced by the preference set. We then perform the minimax merger of these two to generate the final recommendation set Φ, addressing undesirable issues such as an empty set or a large set in the process.

We now describe the details of this design through an implementation in an adapted scenario. This allows us to (1) elucidate our design choices and the various aspects of its functioning in a concrete setting, (2) describe the technologies needed to put it into practice, and (3) reveal the strengths and possible flaws in our theory through pertinent examples.

4.2.2 Database Creation and Preprocessing

A crucial aspect of implementing our framework is the creation of a database, since each scenario has its own peculiarities concerning its content. This is done in the preprocessing unit. The elicitation and recommendation units depend on the structure of the database and not its actual content, and hence can be adapted to another scenario without any further modification.

The Scenario We favour the analysis of user reviews in the characterisation of our scenario since they represent a wealth of honest, subjective information and very real appreciations compared to star ratings, or sponsored promotions. Consisting mainly of positive and negative evaluations, they are ideal for the extraction of boolean values for item features and essential to our system.

With the increasing popularity of online hotel reservations and one-stop shop solutions such as TripAdvisor [2], we settle on the problem of *choosing a hotel* as it is easy to relate to, simple to understand and yet complex enough to evaluate the relevance of intelligent text processing techniques.

4.2.2.1 Item Profiling

To generate accurate item profiles, we must ensure that the set of features is truly representative of the items that it characterises. This means that all avail-

2. www.tripadvisor.com

able information about the items must be analysed and correctly categorised into feature-based information.

Corpus Construction: Determining Ω The first step in item profiling, therefore, is constructing a corpus of all information that can be gathered about the items. This includes objective information, or metadata, such as technical specifications, price, etc.; and subjective information such as user-generated content (reviews, comments, etc.). We therefore create the database for our system from hotel-related metadata and a corpus of user reviews collected from TripAdvisor. We used a technology called web scraping, by means of the open source framework Scrapy[3].

Our algorithm for corpus construction is implemented as a script within the Scrapy framework. This script is designed as a bot that "scrapes/crawls" through the list of Paris hotels in TripAdvisor, collecting metadata and reviews about each individual hotel from its dedicated page on TripAdvisor. The resulting corpus is generated as a collection of hotels, each hotel, in turn, being a collection of all its reviews and metadata. Our corpus contains metadata and reviews from 467 hotels in Paris, 100 reviews each. Thus, in terms of the preference language $\mathcal{L}$ for this scenario, we have Ω = the set of 467 hotels.

Corpus Analysis: Determining Σ The next step is analysing the corpus to determine the set of features formalised as Σ. First, all extracted metadata for every hotel is analysed to finalise the set of objective features that are common to all hotels and will be contained in Σ. Their boolean values will be attributed to their presence and absence respectively.

Next, a set of subjective features is determined by identifying the most mentioned features in the corpus of user-generated content. These are features that can be described positively or negatively in the content, thus allowing the system to compute an aggregate polarised score for each feature for a given hotel. This score, which we call the *p-score*, will be used to assign boolean values to the subjective features.

Based on the extracted hotel-related metadata, we settle on the following list of objective features:

— *Price Range*: divided into 3 classes based on extracted prices for each hotel: Budget, Mid-range and Luxury;
— *Amenities*: Wifi, Non-Smoking, Pets Allowed, Kitchenette, Reduced Mobility Services, Sports Facilities and Luxury Services;
— *Number of Stars*: 5-star, ..., 1-star;

3. www.scrapy.com

— *Location*: 25 distinct locations identified for all 467 hotels in Paris on TripAdvisor [4].

To settle upon a list of subjective features, we look at the Rating Summary provided by TripAdvisor for each hotel. This summary allows users to rate their hotel on 6 features: Location, Sleep Quality, Rooms, Service, Value and Cleanliness. Intuitively, these form a good set of features for preference elicitation. Additionally, we perform a manual analysis of a portion of the corpus to verify this theory and identify other possible features that users may express themselves on.

Since we consider 'location' to be an objective feature (characterised by the geographic address), we do not include it in the final set of subjective features. The feature 'room' is also omitted from this set due to its generic nature. Users mention it in many different contexts (e.g. size, noise, bathroom quality), where it may be positive for a given context and negative for another. This causes a problem aggregating scores at the hotel-specific level, since the aggregation would not be a faithful representation of room quality. We add the feature 'breakfast' to our list as it was frequently mentioned. Thus, our final set of subjective features in $\mathcal{L}$ is:

Cleanliness, Good Breakfast, Good Sleep, Good Service, Good Value.

Note that the feature 'Value' stands for 'value for money' and thus differs from 'Price'. Users may have a positive or negative opinion about the 'value for money' of the object, but not its 'Price', which is an objective feature.

To sum up, we have Σ containing 45 distinct features divided into 5 subcategories. We have four categories of objective features and one category called 'Hotel Quality' which contains all the subjective features.

Polarised Feature Extraction and Aggregation: Generating *Item Profiles* The final step is accurately attributing values to each feature of a given item, to generate an *item profile*. Recalling that we choose to have only binary features in Σ, these values must correspond to `True` or `False`.

For objective features this is straightforward, and corresponds to the presence or absence of the given feature in a given item. We use a combination of language engineering technologies to analyse hotel metadata and automate the assignment of `True` whenever the feature is present, and `False` otherwise. This characterises each objective feature with a boolean value.

For subjective features, it is a longer process: we extract positive and negative aspects of the features by analysing all the user-reviews about each hotel,

4. TripAdvisor provides an exact address *and* a 'location' for each hotel. Extracting the entire list of 'locations' for the 467 hotels and removing duplicates revealed only 25 distinct locations: TripAdvisor itself has categorised all its Paris hotels into only 25 distinct locations.

and aggregate this information at the hotel-specific level to generate p-scores. This is a suitable paradigm in our approach since our recommendations are computed according to the *satisfaction of the user requirements about each individual item feature*. Using an item-ranking approach to recommend the highest rated items, for example, we would not be able to reflect this finer-grained information about each individual feature.

To perform this task, opinion mining/sentiment analysis has proven to be quite effective in extracting information from texts. It provides an added dimension to the information that can be extracted from large textual data: contextual polarity. We use an annotator which is suited to our corpus of hotel reviews (Volkova et al., 2013, 2015).

Since sentiment analysis tools provide annotations at the sentence level and our aim is to assign boolean values to features, we must determine feature-level polarity. We therefore define a text-processing algorithm which first detects features, then combines sentence-level sentiment annotations with feature annotations to generate feature-level annotations.

To detect features within the corpus, we use *Named Entity Recognition* with manually defined gazetteer lists. This is because a feature could be mentioned explicitly in a sentence (e.g. *breakfast* mentioned explicitly in 'the *breakfast* was great'), or it could be implicit in the meaning of this sentence (e.g. *price* mentioned implicitly in 'the hotel is *expensive*'). To generate feature-level annotations, we use a *Named Entity Transducer*, which applies grammar rules to detect annotation patterns.

To implement this algorithm, we use the open-source text engineering framework GATE[5](Cunningham et al., 2002) and create a four-stage application which performs the following tasks sequentially on our corpus:

1. Sentence-level sentiment annotation (Sentiment Annotator): the corpus is processed by sentence, and each sentence is provided a *sentiment label* (negative or positive).
2. Feature annotation (Named Entity Recognition): the corpus is processed by token, and each word that corresponds to a subjective feature in Σ (i.e matches with words that belong to that feature's gazetteer list) is labelled using that feature name. We call this a *feature label* (breakfast, cleanliness, etc).
3. Feature-level sentiment annotation (Named Entity Transducer): we define grammar rules in the transducer to associate feature labels with sentiment labels. Thus every word which is labelled as a feature is given the sentiment label of the sentence it belongs to. This indicates the nature of the feature along with its polarity. We call this a *polarised feature label*.
4. Clean-up: only the polarised feature labels are preserved in the final annotation set. All other labels (sentiment and feature) are deleted.

5. http://gate.ac.uk

In this way, the final annotated corpus reveals all the occurrences of the features in Σ, with an added dimension of polarity [6]. We now know not only when a feature is mentioned, but also if it is mentioned in a positive way or a negative one.

Given that the corpus is a collection of reviews gathered from the 467 hotels in Ω, what remains to be calculated, is the effect this polarity has at a hotel-specific level. We call this the *p-score*.

We use a three-step method to calculate this score:

1. Calculate *P/N ratio*: this is a the ratio of positive to negative mentions. The formula ensures that the value is > 1 (resp. < -1 if there are more positive (resp. negative) mentions.
2. Define a *threshold* for the aggregate neutral point: this is an important step, since we must interpret ratios as boolean values. A high positive (resp. negative) ratio for a given feature indicates that the polarity of all its mentions is predominantly of the same kind. Therefore, the higher the ratio, the easier it is to associate the said feature with a boolean value. When the ratio is closer to 1, it means that there were mixed opinions about the said feature. In this case, assigning a boolean value based on the polarity of the ratio could be misleading. Defining the threshold value, therefore, sets the mark at which the ratio can safely be converted into a boolean value. This threshold is defined manually for each feature, through verification on a test set of reviews for several hotels.
3. Assign *aggregate polarity*, i.e. generate p-score: if the ratio for the feature X is above (resp. below) the threshold value then we have X^+ (resp. X^-). If it is equal, we also have X^+.

Essentially, the P/N ratio [7] is the ratio of positive to negative mentions, which ensures that the value is > 1 (resp. < -1) if there are more positive (resp. negative) mentions. It is mathematically expressed as (assuming that if either count is 0, the formula is not used):

$$\begin{cases} \text{IF positive\_count} \neq 0 < \text{negative\_count , THEN: } (-1) \times \frac{negative\_count}{positive\_count} \\ \text{IF positive\_count} > \text{negative\_count} \neq 0, \text{THEN: } \frac{positive\_count}{negative\_count} \\ \text{ELSE}, 0 \end{cases}$$

This method is applied to every hotel in Ω, to assign a p-score to each of the 5 subjective features characterising it.

6. This annotated corpus is available for download at the following address: `https://seafile.lirmm.fr/d/5aea561d18/`

7. obtained from the open source sentiment analysis tool Semantria (`https://semantria.com`)

Summing-Up Having performed these three steps, all items contained within the recommendation scenario should have a corresponding item profile, ω, to characterise them in terms of our preference language $\mathcal{L}$. Formally, this item profile is *an* n*-tuple, where* n *is the number of features in* Σ*, and each term in the* n*-tuple is either* `True` *or* `False`*. In our hotel-implementation, we have each hotel profile described as a 45-tuple.*

To sum up,

— Σ is the set of item features (45 features in the hotel implementation),
— $\mathrm{Dom}(X)$ for each feature $X \in \Sigma$ is the set $\{\mathrm{True},\ \mathrm{False}\}$, where
— for an objective feature, True (resp. False) corresponds to its presence (resp. absence),
— for a subjective feature, a positive p-score (resp. negative) corresponds to True (resp. False),
— an item profile ω is the result of assigning a truth-value to each feature, and
— Ω is the set of item profiles.

4.2.2.2 Determining Dependencies Between Features

Having generated item profiles, we use the association rule mining technique on these to determine dependencies between item features. Our basic intuition is to deduce correlations between polarised features. Given the large amount of data at our disposal, we perform itemset mining and association rule learning using the open-source data mining tool called WEKA (Hall et al., 2009).

This generates dependency rules between the polarised features that are true for the entire corpus–or the entire set of item profiles. For example, on our entire set of hotel profiles, one of the rules discovered was that *breakfast*$\rightarrow \neg 2 - \mathrm{star} \wedge \neg 1 - \mathrm{star}$.

This information has several uses in the system. During preference elicitation, it reduces the number of interactions with the user and helps the system guide them to express conditional preferences. During its last phase, our recommendation engine resolves situations such as a large recommendation set, or an empty one. This requires, among other things, dependencies between the features to refine the recommendation set or to detect inconsistent preferences, by asking the user to reveal the more important feature between two dependent features.

An unexpected by-product of performing this study was that the quality of the results obtained changed dramatically when the *threshold value* (bias for 0-point in calculating p-score) was modified. This gave us (1) a second and objective means of evaluating our threshold value, since this method is automatic and incorporates the entire corpus for analysis; and (2) a faster, surer way of doing so, since the validity of discovered association rules is easier to evaluate than manually judging the calculated p-score w.r.t a test set of reviews for a given hotel.

The Dataset In WEKA, the dataset is a two-dimensional table: a collection of examples called *instances*, consisting of a number of attributes belonging to a set of attribute types. In our case, each instance is the 45-tuple $h_i \in \Omega$ with Σ_i the set of attribute types. Using this dataset, we mine association rules using the default algorithm provided: *Apriori*.

4.2.3 Elicitation + Recommendation: Our Web-Application for Hotel Recommendation

Using the database generated during the preprocessing phase, we implement the preference elicitation and recommendation phases as a web-application [8]. The preference elicitation unit is designed as a dynamic web page, written primarily in HTML5, formatted using CSS, with dynamic interactions written in Javascript. We use server-side scripting to run our recommendation engine which is integrated into the web-interface using PHP.

In implementing the core computation algorithm, we have simplified the task of interpreting preference statements and items, which theoretically are logical formulae, by processing them as character strings. It is written in Python, to avail of its strong text processing features. Moreover, since all other computations required by the algorithms are in-built features in Python, we avoid using any external libraries to implement our AI. This allows for simple and efficient integration into the web-interface.

Let us now look into the design aspects of this web-application.

4.2.3.1 Preference Elicitation Unit

At the theoretical level, after preprocessing, the preference language $\mathcal{L}$ is completely instantiated for the given recommendation scenario. The next step is to generate the preference set $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes}$ by eliciting the user's strong, optimistic, pessimistic, and conditional/defeasible preferences about the item features, many of which could be implicit. This is performed in the web-application by implementing an interactive preference elicitation protocol.

The theoretical steps that the elicitation protocol must follow are summarised as:

1. Elicit comparative preference statements,

8. http://www.lirmm.fr/~patel/PhD/index.html

2. Identify strong vs not strong preferences,
3. For each elicited statement, determine optimistic/pessimistic bias,
4. Repeat for conditional/defeasible preferences.
5. Result: two preference sets–one containing all preferences with optimistic bias and the other with pessimistic bias.

We now take up each theoretical step of the elicitation protocol and explain how it is accomplished through user interactions on the façade and interpretations under the hood. We shall use the following walk-through example to elucidate.

> Walk-through Example: Anita would like a budget hotel in Paris, she's a stickler for cleanliness and loves her beauty sleep. She doesn't really care for a good breakfast, as she prefers her morning coffee at charming café terraces while she discovers the city. Wifi, non-smoking rooms and not having pets around would be a plus for her.

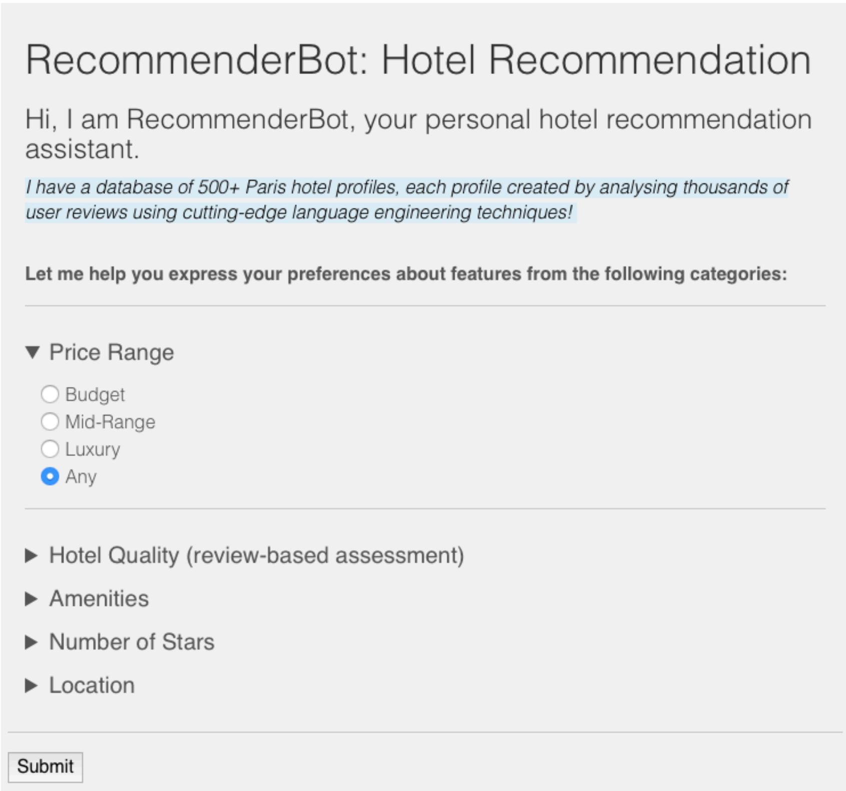


Figure 4.1 – Graphical Interface: Homepage

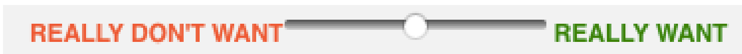
Steps 1 and 2: Eliciting and identifying strong vs not strong comparative preference statements When Anita fires up our web-application, she is greeted with the homepage (Fig. 4.1) which presents her with a list of all the features in Ω, i.e. all hotel features. We display the 45 features under their different categories: price range, hotel quality, amenities, number of stars, location. Each category name is visible, and if Anita seeks to express her preference about a feature in a given category, she may expand the list to reveal the features it contains. In Figure 4.1, we see that Anita is about to express her preference about the price category.

The first interaction which would be elicited from Anita is designed to allow her to express her preferences about each individual feature by degree of strength. In terms of comparative preference statements, this would allow the system to identify her strong preferences.

To explain what happens on the façade, we first define the notion of feature importance. We use this to identify those features which the user would certainly want (or not want) in the set of preferred items, i.e. about which they have a preference.

Definition 20 (Feature Importance)**.** Let $X \in \Sigma$. We say that X is an *important* feature to indicate that X must appear in at least one comparative preference statement in the preference set $\mathcal{P}_{\triangleright}$.

In its most general form, to elicit the user's importance for each feature in Σ, we have devised a graphical interaction which allows them to quantify this importance. We use the following *slider*:



This slider contains a 5-point graded scale: extreme left, left, mid-point, right, extreme right, indicating the varying degrees of importance that the user may accord to having (right side), or not having (left side) this feature. The default position is the mid-point, indicating indifference. This additionally allows the user to express a strong preference about *not* wanting a certain feature.

This is a description of this step in its most general form, as it should be considered before implementation. When deploying this step in a specific implementation, several adaptations can be made, as we can see for example in the case of our own implementation in Figure 4.1: the elicitation for the 'price range' category does not use a slider-based interaction. This adaptation is appropriate for this category, as the features are mutually exclusive.

Getting back to the general from, under the hood, we interpret this graded importance of features as the *strength* of preference about having or not having them: the 5-point scale allows us to distinguish between high importance (= strong preference), low importance (= preference without an associated semantics) and no importance (= no preference) for the presence or absence of a given feature. This graphical interaction is defined to mirror our linguistic analysis of strong vs not-strong preferences (see Table 2.9 on page 47): strong preferences can be identified by the presence of 'enhancer'/'qualifier high' keywords (e.g. superlatives).

Thus, the following comparative preference statements can be formulated, based on the slider configuration of a given feature X:

— slider on the right side: $X \triangleright \neg X$,
— slider on the extreme right side: $X \triangleright_{st} \neg X$,
— slider on the left side: $\neg X \triangleright X$,
— slider on the extreme left side: $\neg X \triangleright_{st} X$,
— slider in the middle: No preference.

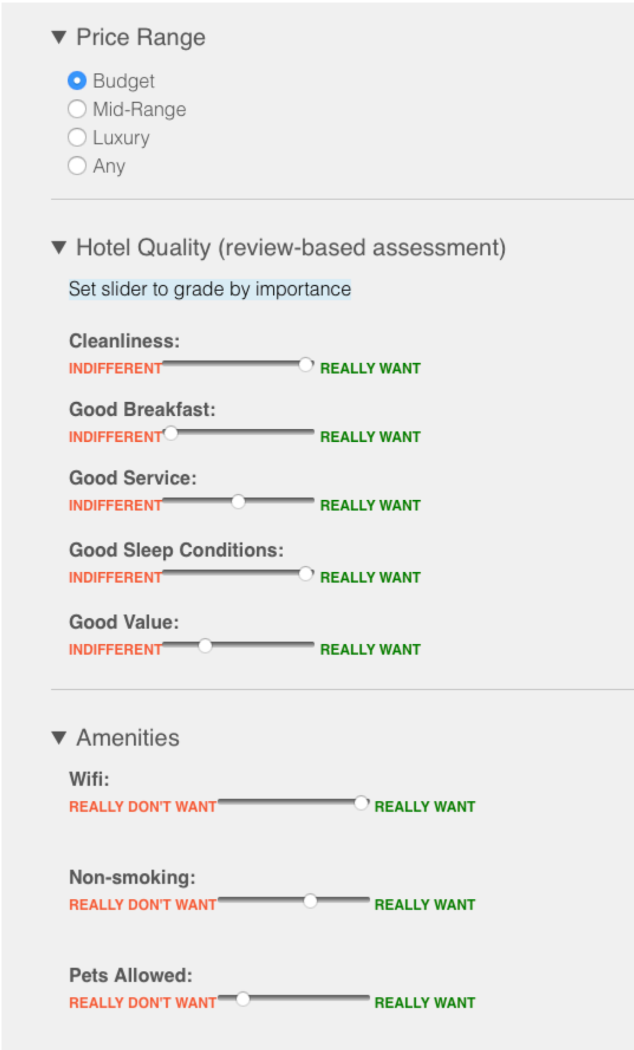


Figure 4.2 – Walk-Through Example: Identifying Strong vs Not Strong Preferences

Now that we know how the system interprets the user's preferences, let us see how this applies to our walk-through example. Figure 4.2 on the preceding page shows how Anita would set her preferences in our graphical interface.

This leads to the following deductions by the system:

1. Budget $\rhd$ ¬ Budget
2. Cleanliness $\rhd_{st}$ ¬ Cleanliness
3. Good Service $\rhd$ ¬ Good Service
4. Good Sleep Conditions $\rhd_{st}$ ¬ Good Sleep Conditions
5. Good Value $\rhd$ ¬ Good Value
6. Wifi $\rhd_{st}$ ¬ Wifi
7. Non-smoking $\rhd$ ¬ Non-smoking
8. ¬ Pets Allowed $\rhd_{st}$ Pets Allowed

Note that the slider for features in the 'Hotel Quality' category has been adapted to a 3-point slider. This slider is restricted to expressing only features that are desired, since it makes no sense for expressly desiring bad quality!

To sum up, this first series of interactions allows the system to elicit comparative preference statements that can be associated with strong semantics when the slider indicates extreme positions. Next, setting the slider for a given feature prompts the system to launch the next interaction for this feature, which would allow it to associate optimistic or pessimistic semantics to all elicited preferences (strong and otherwise).

Step 3: Determine optimistic/pessimistic bias for each elicited statement As we indicated earlier (§4.1.4), the elicitation of optimistic and pessimistic semantics requires additional interactions with the user. Thus, as soon as a slider is moved for a given feature (i.e. a comparative preference statement (strong or otherwise) has been elicited), the user is asked an additional question. This question is designed to elicit the choice between optimistic and pessimistic semantics.

On the façade, the user is questioned about their choice in case of a compromise. Depending on whether a given feature is wanted (slider on the right side) or not wanted (on the left side of the 5-point slider), this question differs as following:

— slider on the right side (feature wanted): the user is asked whether the *absence* of the feature is *acceptable* or not,
— slider on the left side (feature not wanted): the user is asked whether the *presence* of the feature is acceptable or not.
— slider in the middle: No question asked.

We design a graphical interaction to elicit this answer as shown in Figure 4.3:

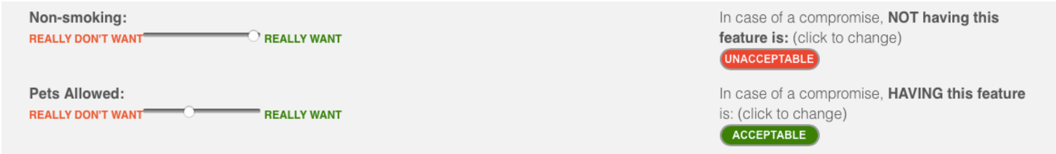


Figure 4.3 – Graphical Interface: Eliciting Optimistic vs Pessimistic Semantics

Under the hood, this allows the system to determine an optimistic/pessimistic bias on the elicited preferences. This deduction is based on the bipolar treatment of preferences which is also corroborated by our linguistic analysis about these semantics (c.f. Table 2.9 on page 47).

It has been shown that optimistic (resp. pessimistic) semantics behave in the same way as negative (resp. positive) preferences (see Kaci (2011, §3.6 on p.76)); where negative (resp. positive) preferences are expressed over a set of outcomes as those which are "(more or less) tolerable or unacceptable" (resp. "really satisfactory"). Thus preferences following optimistic (resp. pessimistic) semantics correspond to *constraints* that *must* be respected (resp. *wishes* which should be satisfied *as best as possible*). The distinction between optimistic and pessimistic preferences lies therefore in their satisfaction: the former "must" while the latter should "as best as possible". It is this aspect that we elicit in our interaction with the user.

By eliciting the user's tolerance for the *opposite* state of the feature X, the answer "unacceptable" indicates that all items in $\mathrm{Mod}(\neg X)$ are unacceptable in the recommendation set Φ, i.e. the preference "must" be respected. Conversely, the answer "acceptable" accepts items belonging to $\mathrm{Mod}(\neg X)$ in Φ, i.e. the preference should be satisfied "as best as possible". This also corroborates with the definitions of optimistic and pessimistic semantics (repeated for ease of reference in § 4.1.3 on page 100). We set the default answer for the user as "unacceptable".

To sum up, all elicited comparative preference statements for a given feature X are interpreted as *optimistic* preferences when the additional question for the feature involved has been answered as *"unacceptable"*, and pessimistic otherwise.

The keen observer would note at this point that this interaction is elicited for *all* features for which the user expressed a preference. This includes those for which the interpreted preference was already associated with strong semantics. Given that the aim of the elicitation unit is to interpret user preferences as comparative preference statements with associated semantics, why then was it necessary to elicit this bias for strong preferences? The answer to this question is provided in §4.2.3.2. For now, we shall merely say that the optimistic/pessimistic bias for strong preferences allows them to be treated in the same way as optimistic/pessimistic preferences. This simplifies the computation of a recommendation set, all the while maintaining the expressivity of strong preferences.

Treating the preferences elicited in steps 1 and 2 through a single interaction in this bipolar way therefore allows us, so to speak, to kill two birds with one stone: (1) we identify a semantics for preferences where no semantics was previously associated, thereby refining $X \rhd \neg X$-statements as $X \rhd_{opt} \neg X$- or $X \rhd_{pes} \neg X$-statements; and (2) we indicate an optimistic/pessimistic bias for strong preferences. Thus, we formally elicit four semantics that can be associated with comparative preference statements. We have: $X \rhd_S \neg X$ for $S \in \{stO, stP, opt, pes\}$

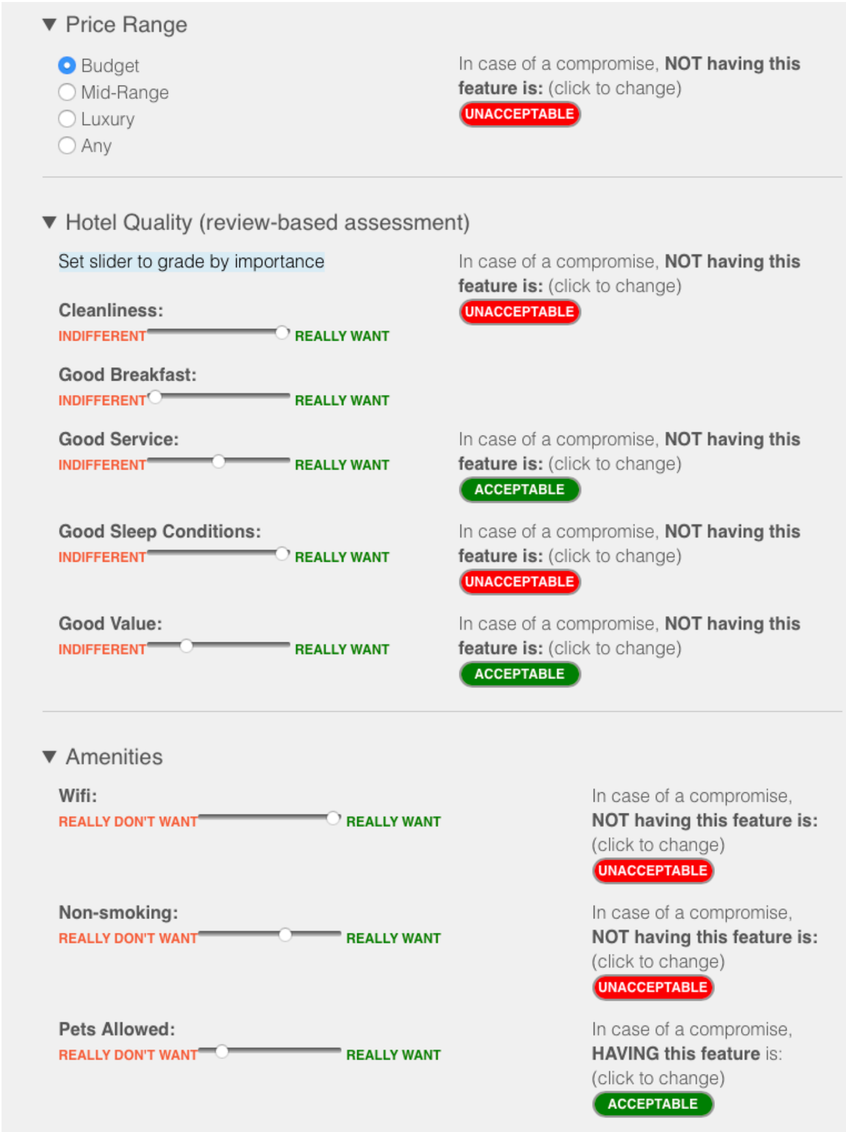


Figure 4.4 – Walk-Through Example: Eliciting Optimistic vs Pessimistic Semantics

Continuing with our example with Anita's preferences in the hotel recommendation scenario, Figure 4.4 shows how she could respond to the second interaction.

The system then interprets these answers as:

1. Budget $\rhd_{opt}$ $\neg$ Budget
2. Cleanliness $\rhd_{stO}$ $\neg$ Cleanliness
3. Good Service $\rhd_{pes}$ $\neg$ Good Service
4. Good Sleep Conditions $\rhd_{stO}$ $\neg$ Good Sleep Conditions
5. Good Value $\rhd_{pes}$ $\neg$ Good Value
6. Wifi $\rhd_{stO}$ $\neg$ Wifi
7. Non-smoking $\rhd_{opt}$ $\neg$ Non-smoking
8. $\neg$ Pets Allowed $\rhd_{stP}$ Pets Allowed

Step 4: Repeat for Conditional/Defeasible Preferences We design the elicitation of conditional preferences using the dependencies discovered during preprocessing (§4.2.2.2), and the current elicited preferences. Dependencies involving features from the elicited preferences could require the user to make a conditional choice: for example, if the user wants a good breakfast *and* a 2-star hotel, the following rule *breakfast*$\rightarrow \neg 2-\mathtt{star} \wedge \neg 1-\mathtt{star}$ would force them to anticipate situations where one of the two is not satisfied. We therefore seek to elicit conditional preferences for the anticipated situations where (1) the breakfast is not good, and (2) hotels have more than 2 stars.

To ensure that the user expresses the conditional preference correctly, a drop-down list is provided to suggest the relevant features. To continue with our example, the user would then be presented with a fill-in-the-blanks styled statement: *"if [X] is not possible, then I'd want [Y]"*, where X is the drop-down set of the features "breakfast, 1-star and 2-star" and Y is the set of all features Σ. The user is allowed to express as many preferences of this kind as they like.

Under the hood, since we deal with binary features, these preferences are formally expressed as $\neg X \wedge Y \rhd \neg X \wedge \neg Y$.

To elicit defeasible preferences, the user is presented with the set of comparative preference statements which have already been deduced, and is asked to indicate when there are specific situations where the preference might be reversed. They provide their answers by selecting the situation from a drop-down set of the binary features.

Our hotel-based recommender is a prototype implementation of this framework to test the feasibility of the entire system design. We have omitted this last step in our implementation due to its cognitive and computational demands on the user and system respectively. This implementation shall, nevertheless, allow us to evaluate all other interactive aspects, along with the performance of our reasoning engine. As we shall see, the results of our feasibility study show that it is necessary to anticipate trade-offs that the user may be forced to make.

Consequently, this step cannot be avoided in a complete implementation of the proposed framework.

Step 5: The Result The information collected through these interactions allows the system to generate the preference set $\mathcal{P}_{\triangleright} = \mathcal{P}_{st} \uplus \mathcal{P}_{opt} \uplus \mathcal{P}_{pes}$. We now look at the core of the recommendation system: the recommendation engine.

4.2.3.2 Recommendation Engine

The recommendation engine in our web-based application takes the preference set generated by the preference elicitation unit and computes a recommendation set for the user. It performs four tasks: (1) compute $\succeq_{opt}$ and $\succeq_{pes}$; (2) resolve situations of empty preferred sets; (3) compute the minimax merger; and (4) resolve situations of very large preferred sets to determine the final recommendation set Φ.

Now, recall that while describing the design of our elicitation unit, we said that it was necessary to elicit an optimistic/pessimistic bias for strong preferences, indicating that our reasons would be clarified when we explain the workings of our reasoning engine. It is time to do so now.

How does the optimistic/pessimistic bias help the system handle strong preferences? Let us recall that there are two distinguished models for $\mathcal{P}_{st}$: the least- and most-specific models. Recalling also from Def. 19 on page 105 that the recommendation set Φ corresponds to the set of un-dominated items $\omega \in \Omega$, we can infer that this set would vary depending on the distinguished model we compute. If $(E_1, ..., E_n)$ is the least-specific model, and $(E'_1, ..., E'_{n'})$ is the most-specific one, then Φ would either be E_1 or E'_1, depending on the model we wish to compute. Algorithms 1 and 2 on page 71 compute these two models respectively, but how to determine which, based on the user's preference?

This is where the optimistic/pessimistic bias comes into play. An optimistic (resp. pessimistic) bias for a given strong preference indicates that we seek to compute the *least-specific* (resp. *most-specific*) model for the preference. Algorithms 1 and 2 can thus be respectively used for the purpose.

However, let us keep in mind that we seek to recommend only the undominated items, i.e. E_1 or E'_1 respectively. We note here that it can be verified by construction that the algorithm for optimistic (resp. pessimistic) preferences *does* compute E_1 (resp. E'_1) (though not the subsequent ones in the ordered partition). We can therefore conclude that Algorithms 5 and 6 on page 74 *suffice for generating the recommendation set* Φ *in the design of our system*.

Step 1: computing $\succeq_{\mathtt{opt}}$ and $\succeq_{\mathtt{pes}}$ We therefore consider all strong preferences with an optimistic (resp. pessimistic) bias as optimistic (resp. pessimistic) preferences and sort them respectively into $\mathcal{P}_{\mathtt{opt}}$ and $\mathcal{P}_{\mathtt{pes}}$. Consequently, $\mathcal{P}_{\triangleright} = \mathcal{P}_{\mathtt{opt}} \uplus \mathcal{P}_{\mathtt{pes}}$ and we can perform our first task of computing $\succeq_{\mathtt{opt}}$ and $\succeq_{\mathtt{pes}}$ by implementing Algorithms 5 and 6 respectively.

Let us take up our walk-through example with Anita to see what implementing these algorithms in the hotel recommendation scenario produces for her. From her elicited preferences, we have:

$\mathcal{P}_{\mathtt{opt}} = \{$
Budget $\triangleright_{\mathtt{opt}} \neg$ *Budget*
Clean $\triangleright_{\mathtt{stO}} \neg$ *Clean*
Good Sleep $\triangleright_{\mathtt{stO}} \neg$ *Good Sleep*
Wifi $\triangleright_{\mathtt{stO}} \neg$ *Wifi*
Non-smoking $\triangleright_{\mathtt{opt}} \neg$ *Non-smoking*
$\}$

$\mathcal{P}_{\mathtt{pes}} = \{$
Good Service $\triangleright_{\mathtt{pes}} \neg$ *Good Service*
Good Value $\triangleright_{\mathtt{pes}} \neg$ *Good Value*
$\neg$ *Pets Allowed* $\triangleright_{\mathtt{stP}}$ *Pets Allowed*
$\}$

The five preferences in the optimistic set are taken together and Algorithm 5 runs under the hood to compute E_1 consisting of 8 hotels from the total set of 467 hotels in Ω. Next, the three pessimistic preferences are taken together and Algorithm 6 generates E_1', which consists of 330 hotels from Ω. The following are the 8 hotels in E_1:

Ermitage Hotel Sacre-Coeur,
Fred Hotel,
Grand Hotel Dore,
Hotel Albe Bastille,
Hotel Apollon Montparnasse,
Hotel Devillas,
Hotel Palym,
La Maison Montparnasse.

Step 2: Resolve situations of empty preferred sets Once computed, if either partition contains an empty preferred set, it is an indication of (1) inconsistent preferences or (2) the absence of a hotel that satisfies the preferred configuration. The latter can easily be verified by computing this configuration in an ideal situation where all hotel configurations are possible (i.e. no integrity constraints). Resolving the former leads to the second task of the engine. The design of our elicitation unit prevents the user from expressing contradictory preferences for a given feature. Thus the only inconsistency can come from incompatible preferences. For this, we have two possible solutions:

1. Identifying inconsistent preferences based on the dependencies discovered during preprocessing (§4.2.2.2), questioning the user on the relative importance of the features involved and eliminating preferences involving the less important one(s) from $\mathcal{P}_{\triangleright}$ (e.g. Given the discovered dependency *breakfast*$\rightarrow$ $\neg$*service*, if $\mathcal{P}_{\triangleright}$ contains *breakfast* $\triangleright\neg$*breakfast* and *service* $\triangleright\neg$*service*, then it could lead to an inconsistent set since *breakfast* and *service* cannot both be satisfied. Thus the user is asked to choose the more

important feature between *breakfast* and *service* so that preferences concerning the less important feature can be eliminated.).

2. If the first solution does not resolve the inconsistency, we weaken the user's requirements by keeping only strong preferences and dropping all others. We then re-compute $\succeq_{\mathtt{opt}}$ and $\succeq_{\mathtt{pes}}$.

In theory, these solutions are applied in the event of any of the following three possibilities: (1) empty preferred set in $\succeq_{\mathtt{opt}}$ (2) empty preferred set in $\succeq_{\mathtt{pes}}$ (3) empty preferred set in both. In practice, recalling from the bipolar viewing of preferences that the optimistic partition models constraints which must be respected while the pessimistic one models wishes that can be satisfied as best as possible, we favour the former partition in the recommendation process and chose to disregard situations with an empty preferred set in $\succeq_{\mathtt{pes}}$ to avoid performing the additional steps needed for resolving inconsistencies.

We can see that step 2 is not necessary for Anita's preferences in our walkthrough example, since both E_1 and E_1' have been generated.

Step 3: Compute the Minimax Merger When both recommendation partitions have been computed, the third task is performed: merging them into a single preference relation $\succeq$ using a minimax merger (Def. 18), and consequently generating a final ordered partition of Ω.

We choose this merger since it complies with the bipolar viewing of optimistic and pessimistic preferences: it allows us to respect the user's constraints, and use their wishes to break ties. In this way, the recommendation generated by taking the optimistic preferences is given more importance than the one generated by taking the pessimistic preferences.

Considering for example the extreme case where the user has only one optimistic preference and several pessimistic ones, the final recommended partition will still favour the items satisfying the optimistic preference, since this is a constraint which must be respected, while all the other preferences are, so to say, expendable.

To show how this is reflected in our hotel-based implementation, we take up our long-standing example with Anita's preferences. The result of the minimax merger between the 8 hotels in E_1 and the 330 in E_1' (and consequently the final recommended set Φ) is computed as the following 6 hotels:

Ermitage Hotel Sacre-Coeur,
Fred Hotel,
Grand Hotel Dore,
Hotel Apollon Montparnasse,
Hotel Palym,
La Maison Montparnasse.

The two hotels *Hotel Albe Bastille* and *Hotel Devillas* were originally in E_1 but were omitted from Φ as a result of the minimax merger, because they did not satisfy all of Anita's pessimistic preferences. Figure 4.5 shows the recommendation provided to Anita by our web-application.

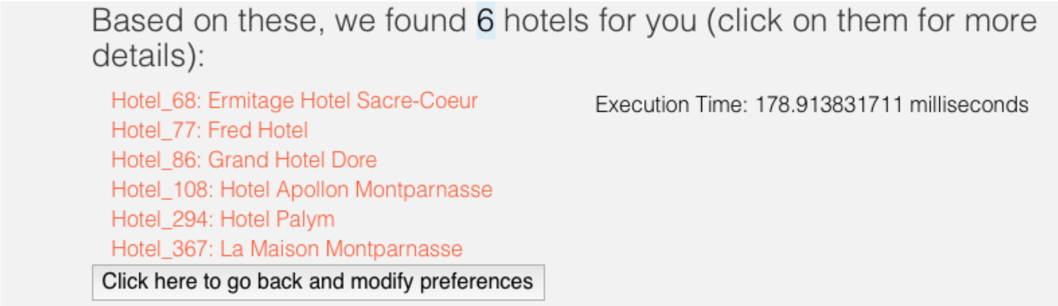


Figure 4.5 – Walk-Through Example: Final Recommendation

Step 4: Resolve situations of large preferred sets The final task is to resolve situations with a large preferred set in $\succeq$. The first means of reducing the number of items proposed is to sort them by quantifiable features. In our hotel-based implementation, we choose these to be 'price' and 'location'. In this way, the list could be truncated to the required size without violating any of the preferences.

It is possible, however, to have situations where this sorting does not suffice: e.g. all recommendations are similarly priced and located according to the user's preferences. We address such situations by questioning the user on the relative importances of features. The dependencies calculated during preprocessing determine the features about which the relative importance is solicited. Thus the preference set is shortened by keeping only the items containing the more important feature(s) (e.g. Given the discovered dependency *breakfast*$\rightarrow$ $\neg$*service*, we know that *breakfast*-items will not contain the feature *service*, and vice versa. Thus the user is asked to choose the more important feature between *breakfast* and *service*, so that the items containing the less important feature can be safely eliminated, without eliminating any items containing the important feature.).

After resolution, the recommendation engine proposes the final recommendation set Φ.

This completes our section on the theoretical design of our proposed framework. We complete this chapter with a feasibility study performed on our hotel-based implementation.

4.3 Results and Observations

Having discussed our theoretical framework for personalised recommendation using comparative preference statements, we now provide results and observations from our hotel-based implementation of each aspect of our proposed framework.

4.3.1 Preprocessing

Corpus Construction The corpus used for preprocessing was constructed by an algorithm designed to scrape hotel-related data from TripAdvisor. The corpus contains a list of hotels in Paris, each hotel containing the following fields: *name, rank, reviews, priceRange, numStars, location, amenities, address, url*. Among the fields, the 'reviews' field is a list of reviews, each review containing the following fields: id, hotelName, title, description.

The following review-related available metadata could be an interesting addition to our system: *starRating, timeOfWriting, timeOfStay, reviewerId, numHelpfulVotes*. This information could be exploited to associate a *confidence-value* to each review, which would then be integrated into the formula used for aggregating individual feature polarities to calculate a hotel-specific feature polarity. For example, a large difference between the time of writing for a given review and time of stay for the user means that there is a chance that hotel has changed since the time of stay; this could be exploited for the final p-score by lowering the confidence-weight of the review in question.

Named Entity Recognition + Sentiment Analysis The following chart (Fig.4.6) shows the average feature occurrence per hotel, as detected by our algorithm for Named Entity Recognition.

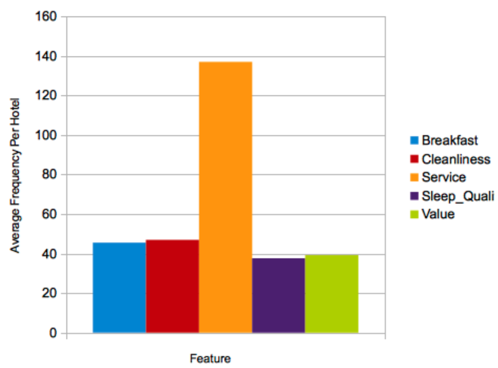


Figure 4.6 – Average Feature Occurrence Per Hotel

Our results depend on the quality of the gazetteer lists (terms to be annotated as a given feature) defined for each feature. For our present purpose, we have defined these manually. The increasing availability of large collaborative knowledge bases for entities means that this manual step will not be necessary for all implementations of our framework, and the accuracy of feature detections can improved.

The following chart (Fig. 4.7) provides a visual representation of 50 hotel profiles (only subjective features). Each horizontal line is a hotel profile composed of its polarised subjective features, differentiated as coloured chunks.

Note that the size of these "feature-chunks" is variable: this is because we represent the features not by their computed boolean values (p-scores), but by their P/N ratio (the size represents the absolute value of this ratio, its position w.r.t the y-axis indicates the polarity). This is because the latter is a more faithful representation of the results of our review-based analysis: the size of the chunks shows the extent to which users had similar opinions about the corresponding features.

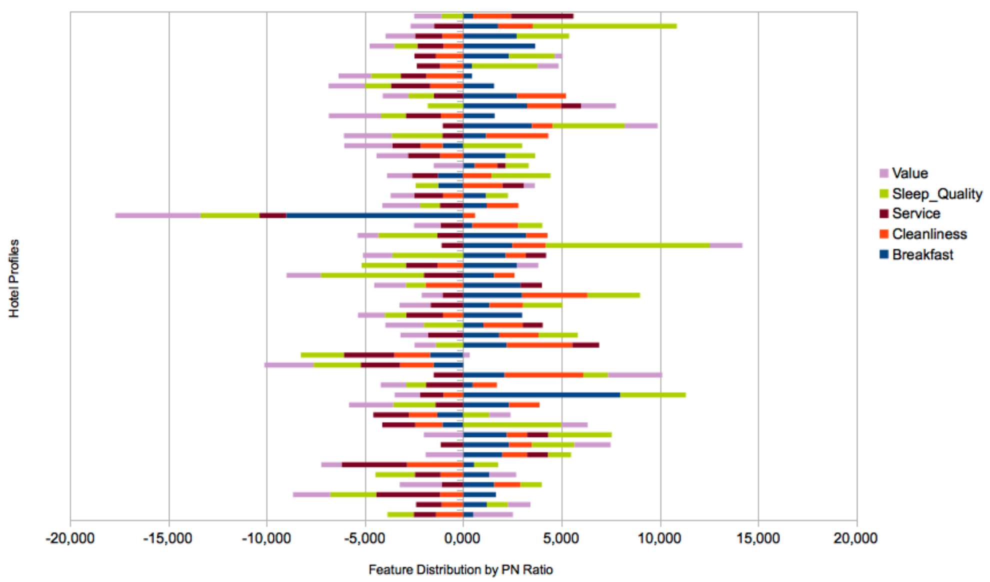


Figure 4.7 – Hotel Profiles for Subjective Features (with frequencies)

Itemset Mining We performed itemset mining to discover dependencies between the hotel features, but have not exploited the discovered rules in our current implementation. The following are some of the rules that we found particularly interesting since they concern features from our review-based analysis, and therefore reflect the dependencies discovered from aggregate user opinions.

1. $\neg$ breakfast $\rightarrow \neg$ value
2. $\neg$ cleanliness $\wedge$ sleep quality $\rightarrow \neg$ service
3. service $\rightarrow$ breakfast
4. service $\wedge$ sleep quality $\wedge$ value $\rightarrow$ breakfast
5. breakfast $\wedge \neg$ cleanliness $\wedge \neg$ sleep quality $\wedge$ value $\rightarrow \neg$ service

6. $\neg$ breakfast $\wedge$ $\neg$ cleanliness $\wedge$ value $\rightarrow$ $\neg$ service
7. cleanliness $\wedge$ $\neg$ service $\wedge$ $\neg$ sleep quality $\rightarrow$ breakfast
8. $\neg$ breakfast $\wedge$ sleep quality $\rightarrow$ $\neg$ service
9. $\neg$ cleanliness $\wedge$ service $\rightarrow$ $\neg$ sleep quality
10. service $\wedge$ sleep quality $\rightarrow$ breakfast
11. cleanliness $\wedge$ service $\wedge$ value $\rightarrow$ breakfast
12. cleanliness $\rightarrow$ $\neg$ service
13. cleanliness $\wedge$ service $\rightarrow$ breakfast $\wedge$ $\neg$ value
14. service $\wedge$ value $\rightarrow$ breakfast
15. $\neg$ cleanliness $\wedge$ $\neg$ sleep quality $\wedge$ value $\rightarrow$ $\neg$ service
16. service $\wedge$ value $\rightarrow$ breakfast
17. $\neg$ breakfast $\wedge$ $\neg$ cleanliness $\rightarrow$ $\neg$ service
18. service $\wedge$ sleep quality $\rightarrow$ breakfast
19. $\neg$ breakfast $\rightarrow$ $\neg$ sleep quality
20. cleanliness $\wedge$ service $\rightarrow$ breakfast
21. $\neg$ cleanliness $\wedge$ value $\rightarrow$ $\neg$ service
22. $\neg$ cleanliness $\rightarrow$ $\neg$ service
23. sleep quality $\rightarrow$ $\neg$ service
24. $\neg$ breakfast $\wedge$ $\neg$ sleep quality $\rightarrow$ $\neg$ service
25. cleanliness $\wedge$ sleep quality $\wedge$ value $\rightarrow$ breakfast

4.3.2 The Web-Application

A proper evaluation of our hotel-based implementation can be performed in a human-subjects experiment designed along the lines of Allen et al. (2015) (see §1.4.2, p.28). To obtain a set of true user preferences would require generating the proper incentive for a group of subjects (as was done by actually offering a restaurant meal in Allen et al. (2015)), or to deploy our framework within an active platform such as TripAdvisor so that users may actually be able to book the hotels that are recommended and comment on them after having stayed there.

An example of the scale at which such a project and consequently its evaluation before deployment have been conducted is the 'Expert Personal Shopper' (XPS) [9], which is a joint collaboration between e-commerce company Fluid [10] and IBM Watson [11] to create a conversational recommender system currently deployed for shopping jackets in partnership with The North Face [12].

At our scale, as an initial step, we assess the feasibility of using our framework with a focus group comprising of 7 volunteers who have no experience in computer science, nor any prior knowledge of what is expected from our web-interface. They performed several tests (by simulating different user-profiles) on the platform and discussed their observations about the platform from a user's perspective. We also performed several tests ourselves to assess the adherence of results of the reasoning engine to those predicted in theory.

We now report the observations from these trial runs and discuss (1) the accessibility and adequacy of our elicitation unit from the user's perspective and (2) the adherence of the computed recommendations to those predicted by the theory on comparative preference statements.

9. This is the tool: `https://www.thenorthface.com/xps`
10. `https://www.fluid.com/software/expertpersonalshopper`
11. `http://www.ibm.com/watson/`
12. `http://www.thenorthface.com`

4.3.2.1 Elicitation Unit

Aside from their own remarks about the experience of using our platform, the members of the focus group were asked to comment more specifically on the following points:

1. *List of features provided:* is it adequate? are there any redundancies? is it too demanding?
2. *Elicitation of Optimistic/Pessimistic Bias:* is what is asked of the user clear? is it too complex?
3. *Ability to express negation in preferences:* is it useful?

1. Selection of Feature List The focus group found the list of features to be generally adequate, although some of them felt that expressing preferences on the number of stars for the hotel was redundant. They argued that given that the users already had a sufficiently detailed set of features to choose from, the services expected from a certain star-rating could already be covered.

In our current implementation, we included all the objective features that could be extracted from the metadata available on TripAdvisor. This list could be refined after a full-scale evaluation of our platform, where the preference sets expressed by all the users participating in the experiment could be analysed to identify the features that were most involved. We could thus verify the focus group's reservations about the feature list and refine our feature list to be better suited to users.

2. Optimistic/Pessimistic Bias The focus group appreciated the possibility of expressing how their preferences should be treated in case of a compromise. In our original test run, we had provided "tolerable" and "not tolerable" (instead of "acceptable" and "unacceptable") as answers to the question to elicit the optimistic/pessimistic bias (c.f. Fig. 4.3 on page 118). We opted to change these answers due to the group's complaint that the double negative in "*not* having a feature is *not* tolerable" was unsettling to them. Again, the utility of this interaction can be quantified in a full-scale evaluation by examining the number of times the users made use of this interaction.

From the theoretical point of view, there is an aspect of this interaction that is cognitively ambiguous. The focus group did not point it out, but it remains pertinent. Our interface essentially elicits two kinds of distinctions through two different interactions: (1) strong vs not-strong and (2) optimistic vs pessimistic. Since we elicit our second distinction by applying a bipolar viewing of preferences, there is an ambiguity that remains unresolved when combining both interactions from the user's point of view: that between optimistic preferences and strong preferences with a pessimistic bias. Let us consider the last two preferences in our walk-through example with Anita (Fig. 4.4 on page 119). The following image is the relevant portion from Fig.4.4.

The first preference is interpreted as an optimistic one, and the second is a strong one with a pessimistic bias. In cognitive terms, the distinction between the two preferences seems to be quite vague: which of these preferences is actually more important for the user? The former or the latter? According to the design of our system, the former is given more importance. But is it truly the case from the user's perspective?

3. Negation in Preferences The group showed a particular interest in this aspect of our graphical interface. They acknowledged that they recognised this to be something that could be useful not only for the current implementation, but for several other scenarios.

A general remark from the focus group was the suggestion that system prompts that would automatically suggest trade-offs would be appreciated by users for better preference elicitation. Typically, when the user expresses a certain preference about a given feature, the system should alert them immediately about the trade-offs they may be forced to make in this situation. This remark is both reassuring and stimulating: we have theoretically foreseen a process by which the system could perform such a feat–our elicitation of conditional preferences exploiting integrity constraints–but we were unsure about the cognitive exigencies this would impose on the user. This general remark is therefore reassuring, since it is clear that it is not an exigency but a necessity. It is moreover stimulating, because we already know that the current implementation of our reasoning engine is already capable of handling such preferences.

4.3.2.2 Recommendation Engine

The results of all the test runs involved in this feasibility study were logged on the server running our web-application. We were therefore able to compute the average execution time for the 156 test runs recorded: `186.1929618395 milliseconds`.

In a further investigation of the results recorded, we observed that:

1. All computed partitions $\succeq_{opt}$ and $\succeq_{pes}$ had a maximum of 2 computed classes
2. Within these, we consistently found $|E_1| < |E_1'|$

These two observations are corroborated by the theory behind comparative preference statements and additionally substantiate the treatment of optimistic and pessimistic preferences as bipolar preferences.

Observation 1: Since we have restricted the implementation of our framework to only elicit comparative preference statements of the form $X \triangleright \neg X$ for a given feature X and since all features have a boolean domain, every item $\omega \in \Omega$ either satisfies the preference or not, i.e. there can be no item for which a preference is not applicable. This is why the algorithms that compute $\succeq_{opt}$ and $\succeq_{pes}$ always generate an ordered partition of only 2 classes. Note, however, that this is not the case when the elicited preference set contains conditional preferences. In the latter case, the system would be able to provide a stratification of recommendations to the user, along with the preferences satisfied by each stratum.

An interesting side note concerning this point is that we found a bug hidden in our code which had escaped our attention when we tested our algorithms against well-known examples. When one of our test runs revealed an ordered partition with 4 classes, we knew there had to be something wrong with our code!

Observation 2: The engine consistently generates fewer recommendations when users choose to consider their preferences as constraints (optimistic semantics) than when they allow them to be satisfied as best as possible (pessimistic semantics). This corroborates the bipolar viewing of preferences, and therefore indicates that our interaction for the elicitation of these two semantics has been successful.

We also performed tests to assess and confirm that the reasoning engine adheres to the theory behind the addition of bipolar preferences: additional optimistic preferences may only reduce the number of preferred items while additional pessimistic preferences may only increase the number of preferred items.

To sum up, our results of the hotel-based implementation and its accompanying feasibility study shows the engineering it takes to adapt our proposed framework to a given scenario, and confirms that it works in a competent time frame, with a reasonable amount of user interaction.

Conclusion

In this chapter, we applied the results of the previous chapters to propose the theoretical framework and an accompanying implementation of a recommender system using a preference logic based AI. We addressed a user for personalised decision support by eliciting their current preferences and providing a recommendation based exclusively on these preferences. Relying both on statistically-driven AI for polarised feature detection and logic-based AI gleaned from theoretical studies about reasoning with preferences, our system consists of (1) a preprocessing unit, (2) an interactive preference elicitation unit, (3) a preference logic based reasoning engine and (4) a final recommendation module which ensures that the computed recommendation set is satisfactory before providing final results.

Our accompanying implementation is centred on the problem of choosing a hotel, based on an appropriate corpus of hotel reviews that we constructed ourselves. The elicitation and recommendation units are adapted accordingly. In this first application, we did not include the possibility of exploiting dependencies between features in the elicitation and recommendation phases.

We assessed the feasibility of our framework by providing our web-based platform to a focus group of uninitiated users, who volunteered to perform several tests (by simulating different user-profiles) on the platform and discuss their observations. They confirmed that the added expressivity of comparative preference statements is appreciable compared to other decision support tools available today. They also independently pointed out, without prior knowledge that this has been accounted for in our theoretical framework, that system prompts that would automatically suggest trade-offs would be appreciated by users for better preference elicitation.

Our own analysis of the recommendations provided in this experiment shows that, thanks to the preprocessing and elicitation phases, our engine performs in a competent time frame and corroborates with expected results, as predicted by the theory behind comparative preference statements. In particular, the collective results of our experiment concretely substantiate the treatment of optimistic and pessimistic preferences as bipolar preferences: the engine consistently generates fewer recommendations when users choose to consider their preferences as constraints (optimistic semantics) than when they allow them to be satisfied as best as possible (pessimistic semantics). This also indicates that our interaction for the elicitation of these two semantics has been successful.

The present work illustrates an application of some salient theories in preference acquisition and reasoning to recommender systems. Backed by these promising preliminary results, it would be interesting to see this framework implemented in a large-scale evaluation, where user's real needs are catered to. We should be able to integrate the elicitation of conditional and defeasible preferences, as this aspect not been exploited in the current implementation.

We would also like to improve the handling of undesired recommendations by addressing a user's hidden, or implicit priorities between preferences. This would allow us to calculate a preference stratification by priority, and keep only the consistent ones. This could also help resolve our difficulty in eliciting the optimistic and pessimistic bias for strong preferences: namely the priority the user would have between optimistic but not strong preferences and strong preferences with a pessimistic bias, since the cognitive distinction between these two seems ambiguous.

Finally, we would like to explore the possibility of integrating the results of our postulate-based analysis to optimise the selection of comparative preference statements in the final preference set: inferring new preferences based on elicited preferences by exploiting the properties of composition/decomposition of the preference statements.

With the increasing development of AI bots to provide personalised assistance to users, we believe that our work is a first step towards that aim, and that the proposed techniques could successfully be applied for other ends.

5 Concluding Remarks

Our exploration of preference handling for decision support has been a maritime journey of theories and technologies. Sounding the deep waters of rigorous theoretical analyses and navigating the choppy and fast-moving waves of knowledge-engineering technologies, we seem to have found a passage for the use of preference reasoning in AI for personalised recommendation. It has indeed been fulfilling to see how a thorough theoretical knowledge of a subject is needed for better, more pointed applications with the help of diverse engineering tools. We have merely scratched the surface, exposing the possibilities unlocked with the growing number of technologies capable of handling large quantities of data, and the deeper insights gained from further theoretical research.

We devote this chapter to a discussion of these, by first recapitulating the achievements of the present thesis.

5.1 Summary

Research on preferences in AI has shed light on various ways of tackling problems related to decision making, right from the acquisition of preferences to their formal representation and eventually their proper manipulation. Many of these have successfully been implemented for intelligent 'services' such as recommender systems. Following a recent trend of stepping back and looking at such decision-support systems from the user's point of view, i.e. designing them on the basis of psychological, linguistic and personal considerations, we took up the task of developing an "intelligent" tool which uses comparative preference statements for personalised decision support. We tackled and contributed to different branches of research on preferences in AI: (1) their acquisition, (2) their formal representation and manipulation and (3) their implementation.

Our first contribution concerned the acquisition of preferences which included (1) addressing an existing bottleneck by proposing a method of eliciting user preferences, expressed in natural language (NL), which favours their formal representation and further manipulation; (2) testing the feasibility of this method using a proof of concept experiment, thereby (3) constructing a corpus of preference expressions and an accompanying lexicon of preference terminology.

Our results confirmed that the very nature of turn-by-turn dialogue provides an effectual structure for preference elicitation, something which prose (such as found in a textual corpus) does not fulfil with equal success. The latter contains numerous expressions of opinions, but very few preferences. It is the interactive nature of dialogue which reveals expressions of preferences. Our preference lexicon coupled with our preference templates, the two components of our linguistic framework for identifying preferences, have served to distinguish preference semantics in the NL-expressions elicited.

As a consequence of our crowd-sourcing experiment, we now have a corpus which contains authentic user preferences in natural language corresponding to comparative preference statements and their associated semantics. This provides a concrete link between natural language expressions and research in preferences in artificial intelligence.

The next portion of our study focussed on the theoretical aspects of handling comparative preference statements for decision support. We performed a thorough investigation of the statements and associated semantics, with a rigorous exposition of their formulation and reasoning mechanisms. We followed it up with an analysis w.r.t. some of the basic principles that govern preference logics in general, to support our intuition behind using this formalism.

The results of our study revealed in particular that opportunistic semantics, being the weakest semantics, has bad properties as concerns both, the reasoning mechanisms to compute distinguished preference relations and the postulate-based analysis. Strong, optimistic and pessimistic semantics were found to have interesting properties w.r.t. the composition/decomposition of preferences, the latter two exhibiting a dual behaviour which corroborates existing works on bipolar information. We also found that ceteris paribus semantics does not satisfy many postulates. It only ensures coherence and preference independence and thus does not allow any decomposition/composition.

Our study of the reasoning mechanisms associated with these semantics to compute distinguished preference relations showed these to be a promising approach in designing a framework for personalised decision support. This led us to the final contribution of the thesis: the design and implementation of a framework for personalised recommendation using comparative preference statements.

In the final part of the study, we worked out how all of the above can come together in an intelligent tool, capable of performing personalised decision support. We first designed an interactive module for preference elicitation which uses statistically-driven methods in information retrieval to minimise user interaction, without losing out on expressivity. We then focussed on the design of the core of our system: the reasoning engine. Our reasoning engine computes recommendations for the user, and is entirely based on the theoretical research on comparative preference statements. We completed our study by implementing the proposed framework in a specific scenario, discussing its performance and adherence to the theory's predictions.

Our accompanying implementation was centred on the problem of choosing a hotel, based on an appropriate corpus of hotel reviews that we constructed ourselves. We launched our platform as a web-based crowd-sourcing experiment designed to collect user-satisfaction about the quality of recommendations offered. In this first application, we were unable to include the possibility of exploiting dependencies between features in the elicitation and recommendation phases. In our feasibility study, we used a focus group of uninitiated users, who volunteered to perform several tests (by simulating different user-profiles) on the platform and report their observations. They confirmed that the added expressivity of comparative preference statements is appreciable compared to other decision support tools available today. They also independently pointed out, without prior knowledge that this has been accounted for in our theoretical framework, that system prompts that would automatically suggest trade-offs would be appreciated by users for better preference elicitation.

Our own analysis of the recommendations provided in this experiment showed that, thanks to the preprocessing and elicitation phases, our engine performs in a competent time frame and corroborates with expected results, as predicted by the theory behind comparative preference statements. In particular, the collective results of our experiment concretely substantiated the treatment of optimistic and pessimistic preferences as bipolar preferences: the engine consistently generates fewer recommendations when users choose to consider their preferences as constraints (optimistic semantics) than when they allow them to be satisfied as best as possible (pessimistic semantics). This also indicates that our interaction for the elicitation of these two semantics has been successful.

5.2 Future Directions

Our contributions touch upon and attempt to combine three interesting aspects of AI research, in their relation to preferences: (1) Natural Language

Processing for preference acquisition, (2) Knowledge Representation for compact preference representation and (2) Decision Support for personalised recommendation. Since our own expertise in this could at best be summarised as a rigorous ability to conceive of and achieve this combination at the proof-of-concept phase, and since the ambitions of our project, by its very nature, call for a collaboration with experts from each of these fields, we believe that our work constitutes a necessary first step in achieving this purpose.

From here on, this opens up future directions in each of these aspects. Based on the experience we gained conducting this research, we foresee advances both on the theoretical and the practical fronts. The following are a few possibilities for further research.

Resolving 'problem points' in preference handling In the course of our research applying comparative preference statements to real-world decision problems, we found the following points that need special attention when dealing with users: (1) restricting the preference set to a manageable, yet sufficiently expressive number of preferences, (2) handling the implicit priorities between statements in a preference set (3) the trade-offs that must be considered based on the integrity constraints of the outcome set. These are a few starting points for future research following our work. We discuss these respectively in the following points.

1. Our postulate-based study of comparative preference statements showed how certain semantics had interesting properties as regards the inference of new preferences by composition/decomposition and weakening of preference statements. This form of inference could be exploited to restrict user preferences to a manageable, yet sufficiently expressive number of preferences. It could also be useful in recommendation algorithms that base their results on a previously determined model of user preferences, by inferring new preferences from formerly known ones.
2. We showed that the use of a preference logic in the design of a recommendation engine leads to computing solutions and not predicting possible ones. In our present proposal, we used comparative preference statements, and one of the problems we had was in handling the priorities between statements in a preference set, especially when the computed solutions were unsatisfactory (i.e. too few/many). An interesting future direction in the use of a preference logic would be to explore the combination of conditional logics (our present proposal) with weighted logics in the elicitation phase so that the user's implicit priorities between different preferences can be extracted. Existing literature already shows how comparative preference statements can be quantified using weighted logic distributions. Since the latter also adheres to the bipolar representation of preferences, including this form of reasoning in the

design of the elicitation protocol could allow the proper handling of priorities between preference statements and provide better recommendations.

3. We addressed the problem of trade-offs that must be considered based on the integrity constraints of the outcome set by discovering inter-feature dependencies from review-based information about the outcomes. This was performed in a pre-processing phase. This idea could be pushed further and integrated into the elicitation and recommendation phases, to identify trade-offs based on the current set of preferences. Applying the itemset mining approach not to the entire set of outcomes, but only to those which are relevant for a given preference set, could reveal the necessary trade-offs the user would be forced to make. This requires more computing under the hood during elicitation, but would reduce the chances of having an unsatisfactory recommendation set.

Extending our Contribution to the Semantic Web The development of dedicated terminologies to facilitate Named Entity Recognition is an important task in natural language processing and knowledge representation. We addressed this for the improvement of interactive preference acquisition methods and now have a dedicated preference terminology that corresponds to comparative preference statements. In its current state, it is described using a Formal Concept Analysis-based lattice structure and pattern-recognition rules.

An interesting future direction for this terminology would be to explore how it could be represented using more open/standard representation formats for better applicability and accessibility. Adapting our markup language to those developed for the Semantic Web, such as RDF, OWL or XML, we would be able to associate the entities we describe with existing entities in the Semantic Web, thereby increasing its readability at a much larger level.

Moreover, since these languages have explicitly been designed to represent knowledge, there are several reasoning mechanisms associated with them today. This opens up the chance to explore the adaption of our recommendation algorithm to the semantic web as well.

Dialogue-Based Recommendation using AI Assistant Agents With the advent of smart hand-held devices and the improvement of AI interactions, we now encounter conversational AI assistants, or 'smart bots' for a host of activities. There are dedicated bots that specialise in a particular kind of assistance such as banking (e.g. MyKAI), scheduling (e.g. Amy Ingram from X.ai) etc, and others such as SIRI (Apple), Cortana (Microsoft), Google Now, or Alexa (Amazon) that provide a variety of services.

Our contributions in this thesis point to the possibility of developing a bot that uses our framework for personalised recommendations. Extending the preference elicitation protocol from our linguistic study to conversational or dialogue-based interactions with users would allow us to adapt our framework to the smart bot platform. The added advantage of this conversational platform would be the possibility of exploiting the time a user expects to get an answer to perform complex calculations. We could thus address the problems faced in our current elicitation of preferences by refining the process: we can (1) goad the user to express implicit, ambiguous or inherent preferences by evaluating the context and (2) prompt the user about potential trade-offs during the elicitation phase by simultaneously computing possible results and detecting inconsistencies.

References

Adomavicius, G. and Tuzhilin, A. Toward the Next Generation of Recommender Systems: a Survey of the State of the Art and Possible Extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

Alanazi, E., Mouhoub, M., and Mohammed, B. A preference-aware interactive system for online shopping. *Comput. Inf. Sci.*, 5(6):33, 2012.

Allen, T. CP-nets: From Theory to Practice. In *Algorithmic Decis. Theory*, 2015.

Allen, T., Chen, M., and Goldsmith, J. Beyond theory and data in preference modeling: bringing humans into the loop. In *Algorithmic Decis. Theory*, 2015.

Arrow, K. J. *Social Choice and Individual Values.*, volume 18. Yale university press, 1953.

Balabanović, M. and Shoham, Y. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.

Barberà, S., Bossert, W., and Pattanaik, P. Ranking sets of objects. *Handb. Util. Theory*, 2:893–977, 2004.

Benferhat, S. and Kaci, S. A possibilistic logic handling of strong preferences. In *Jt. 9th {IFSA} World Congr. 20th {NAFIPS} Int. Conf. 2001.*, volume 2, pages 962–967, 2001.

Benferhat, S., Dubois, D., and Prade, H. Possibilistic and standard probabilistic semantics of conditional knowledge bases. *J. Log. Comput.*, 9(6):873–895, 1999.

Benferhat, S., Dubois, D., Kaci, S., and Prade, H. Bipolar Possibilistic Representations. In *Uncertain. Artif. Intell.*, 2002a.

Benferhat, S., Dubois, D., Kaci, S., and Prade, H. Bipolar representation and fusion of preferences on the possibilistic logic framework. In *KR*, pages 421–432, apr 2002b.

Benferhat, S., Dubois, D., and Prade, H. Towards a possibilistic logic handling of preferences. *Appl. Intell.*, 14(3):303–317, 2001.

Bienvenu, M., Lang, J., and Wilson, N. From Preference Logics to Preference Languages, and Back. In *KR*, pages 214–224, 2010.

Boutilier, C., Brafman, R., Hoos, H. H., and Poole, D. Reasoning with conditional ceteris paribus preference statements. In *Uncertain. Artif. Intell.*, pages 71–80. Morgan Kaufmann Publishers Inc., 1999.

Boutilier, C., Bacchus, F., and Brafman, R. {UCP-Networks:} A directed graphical representation of conditional utilities. In *17th Conf. Uncertain. Artif. Intell.*, pages 56–64, 2001.

Boutilier, C., Brafman, R., Domshlak, C., Hoos, H. H., and Poole, D. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.

Bouveret, S., Endriss, U., Lang, J., and Others. Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods. *IJCAI*, pages 67–72, 2009.

Brafman, R. and Domshlak, C. Preference handling-an introductory tutorial. *AI Mag.*, pages 58–86, 2009.

Brafman, R. and Domshlak, C. Introducing variable importance tradeoffs into CP-nets. In *Uncertain. Artif. Intell.*, pages 69–76. Morgan Kaufmann Publishers Inc., 2002.

Braziunas, D. and Boutilier, C. Minimax regret based elicitation of generalized additive utilities. In *Uncertain. Artif. Intell.*, pages 25–32, 2007.

Braziunas, D. and Boutilier, C. Elicitation of Factored Utilities. *{AI} Mag.*, 29(4):79, jan 2009.

Brewka, G., Benferhat, S., and Leberre, D. Qualitative choice logic. *Artif. Intell.*, 157(1-2):203–237, 2004.

Bridge, D., Göker, M. H., McGinty, L., and Smyth, B. Case-based recommender systems. *Knowl. Eng. Rev.*, 20(03):315–320, 2005.

Burke, R. Hybrid recommender systems: Survey and experiments. *User Model. UserAdapted Interact.*, 12(4):331–370, 2002.

Burke, R. and Trewin, S. Knowledge-based recommender systems. *Encycl. Libr. Inf. Syst.*, 69(Supplement 32):175–186, 2000.

Cadilhac, A., Asher, N., Benamara, F., Popescu, V., and Seck, M. Preference Extraction From Negotiation Dialogues. In *ECAI*, volume 242, pages 211–216, 2012.

Charolles, M. Marquages linguistiques et résumé de texte. *Actes du Colloq. Int. Linguist. Résumé texte*, 1990.

Chen, L. and Pu, P. User Evaluation Framework of Recommender Systems. *Proceeding UCERSTI Work. RecSys'10*, pages 14–21, 2010a.

Chen, L. and Pu, P. Experiments on the preference-based organization interface in recommender systems. *ACM Trans. Comput. Interact.*, 17(1):1–33, 2010b.

Chen, L. and Pu, P. Experiments on user experiences with recommender interfaces. *Behav. Inf. Technol.*, 33(4):372–394, 2014.

Chen, L., Chen, G., and Wang, F. Recommender systems based on user reviews: the state of the art. *User Model. User - Adapt. Interact.*, 25(2):99–154, 2015.

Chen, L. and Pu, P. Survey of Preference Elicitation Methods. Technical report, EPFL, 2004.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL*, pages 168–175, 2002.

Das, M. and Morales, G. D. F. Learning to question: leveraging user preferences for shopping advice. In *ACM SIGKDD*, pages 203–211, 2013.

Domshlak, C. A snapshot on reasoning with qualitative preference statements in AI. In *Prefer. Similarities*, pages 265–282. Springer, 2008.

Domshlak, C., Hüllermeier, E., Kaci, S., and Prade, H. Preferences in AI: An overview. *Artif. Intell.*, 175(7-8):1037–1052, 2011.

Doyle, J. and Thomason, R. H. Background to Qualitative Decision Theory. *AI Mag.*, 20(2):55–68, 1999.

Doyle, J. and Wellman, M. Representing preferences as Ceteris Paribus comparatives. *Work. Notes AAAI Spring Symp. Decis. Plan.*, 1994.

Dubois, D. and Prade, H. An introduction to bipolar representations of information and preference. *Int. J. Intell. Syst.*, 23(8):866–877, 2008.

Dubois, D., Lang, J., and Prade, H. Possibilistic logic. *Handb. Log. Artif. Intell. Log. Program. (vol. 3)*, pages 439–513, 1994.

Dubois, D., Godo, L., and Prade, H. Weighted logics for artificial intelligence – an introductory discussion. *Int. J. Approx. Reason.*, 55(9):1819–1829, 2014.

Dubois, D., Kaci, S., and Prade, H. Ordinal and absolute representations of positive information in possibilistic logic. In *NMR*, 2004.

EARMAN, J. AND ROBERTS, J. Ceteris paribus, there is no problem of provisos. *Synthese*, 118(3):439–478, 1999.

EKSTRAND, M. AND KLUVER, D. Letting Users Choose Recommender Algorithms: An Experimental Study. In *ACM Conf. Recomm. Syst.*, pages 11–18, 2015.

FISHBURN, P. Preference structures and their numerical representations. *Theor. Comput. Sci.*, 217(98):359–383, 1999.

FISHBURN, P. C. *Utility Theory for Decision Making*. Wiley, New York, 1970.

FREUND, M. On the revision of preferences and rational inference processes. *Artif. Intell.*, 152(1):105–137, 2004.

FÜRNKRANZ, J. AND HÜLLERMEIER, E. *Preference Learning*. Springer, 2010.

GOLDSMITH, J. AND JUNKER, U. Preference Handling for Artificial Intelligence, jan 2009.

GONZALES, C. AND PERNY, P. {GAI} networks for utility elicitation. *9th Int. Conf. Princ. Knowl. Represent. Reason.*, 4:224–233, 2004.

GUERIN, J. T., ALLEN, T. E., AND GOLDSMITH, J. Learning CP-net Preferences Online from User Queries. *Algorithmic Decis. Theory, LNCS*, 8176:208—–220, 2013.

HADDAWY, P. AND HANKS, S. Representations for Decision-Theoretic Planning: Utility Functions for Deadline Goals. *KR*, 1992.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA data mining software. In *ACM SIGKDD Explor.*, volume 11, pages 10–18, nov 2009.

HALLDÉN, S. *On the Logic of "better"*, volume 2. C. W. K. Gleerup, Lund, and Copenhagen : Ejnar Munksgaard (Library of Theoria), 1957.

HANSSON, S. O. What is ceteris paribus preference? *J. Philos. Log.*, 25(3):307–332, 1996.

HANSSON, S. O. *The structure of values and norms*. Cambridge University Press, 2001.

HU, M. AND LIU, B. Mining and summarizing customer reviews. In *ACM SIGKDD*, volume 04, page 168, 2004.

JINDAL, N. AND LIU, B. Identifying comparative sentences in text documents. In *ACM SIGIR*, page 244, 2006.

KACI, S. *Working with Preferences: Less Is More*. Springer, 2011.

KACI, S. A postulate-based analysis of comparative preference statements. In *FLAIRS*, 2012a.

KACI, S. Characterization of Positive and Negative Information in Comparative Preference Representation. In *ECAI*, pages 450–455, 2012b.

KACI, S. AND VAN DER TORRE, L. Merging optimistic and pessimistic preferences. In *Inf. Fusion*, 2006.

KACI, S. AND VAN DER TORRE, L. Reasoning with various kinds of preferences: Logic, non-monotonicity, and algorithms. *Ann. Oper. Res.*, 163(1):89–114, mar 2008.

KACI, S., PATEL, N., AND PRINCE, V. From {NL} Preference Expressions to Comparative Preference Statements: {A} Preliminary Study in Eliciting Preferences for Customised Decision Support. In *ICTAI*, pages 591–598, 2014.

KANTOR, P. B., ROKACH, L., RICCI, F., AND SHAPIRA, B. *Recommender systems handbook*. Springer, 2011.

KEENEY, R. L. AND RAIFFA, H. Decisions with multiple objectives – preferences and value tradeoffs. *John Wiley sons*, page Cambridge University Press, 1976.

KNIJNENBURG, B. P., WILLEMSEN, M. C., GANTNER, Z., SONCU, H., AND NEWELL, C. Explaining the user experience of recommender systems. *User Model. User-Adapted Interact.*, 22(4-5):441–504, 2012.

KONSTAN, J. AND RIEDL, J. Recommender systems: from algorithms to user experience. *User Model. User-adapt. Interact.*, 22(1-2):101–123, 2012.

KOREN, Y. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53 (4):89–97, 2010.

KOREN, Y., BELL, R., AND VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer (Long. Beach. Calif).*, 8:30–37, 2009.

KORICHE, F. AND ZANUTTINI, B. Learning conditional preference networks. *Artif. Intell.*, 174(11):685–703, 2010.

KRANTZ, D. H., LUCE, R. D., SUPPES, P., AND TVERSKY, A. Foundation of Measurement - Additive and Polynomial Representation, Volume 1. *Acad. Press. New York*, I, 1971.

KRAUS, S., LEHMANN, D., AND MAGIDOR, M. Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artif. Intell.*, 44:167–207, 1990.

LANG, J. AND MENGIN, J. The complexity of learning separable ceteris paribus preferences. In *IJCAI*, pages 848–853, 2009.

Liu, J., Yao, Z., Xiong, Y., Liu, W., and Wu, C. Learning conditional preference network from noisy samples using hypothesis testing. *Knowledge-Based Syst.*, 40:7–16, 2013.

Liu, W., Wu, C., Feng, B., and Liu, J. Conditional preference in recommender systems. *Expert Syst. Appl.*, 42(2):774–788, 2015.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. Recommender System Application Developments: A Survey. *Decis. Support Syst.*, 74:12–32, 2015.

Lucas, N. Syntaxe du paragraphe dans les articles scientifiques en japonais et en fran{\textbackslash}ccais. *Parcour. Linguist. discours spécialisés*, pages 249–262, 1993.

McCarthy, J. Circumscription—A form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980.

McDermott, D. and Doyle, J. Non-monotonic logic I. *Artif. Intell.*, 13(1-2): 41–72, 1980.

McSherry, D. Retrieval failure and recovery in recommender systems. *Artif. Intell. Rev.*, 24(3-4):319–338, 2005.

Moeschler, J. *Modélisation du dialogue: représentation de l'inférence argumentative*. Hermes, 1989.

Musat, C., Liang, Y., and Faltings, B. Recommendation Using Textual Opinions. *IJCAI*, pages 2684–2690, 2013.

Nunes, I., Miles, S., Luck, M., Barbosa, S., and Lucena, C. Decision making with natural language based preferences and psychology-inspired heuristics. *Eng. Appl. Artif. Intell.*, 42:16–35, 2015.

Öztürk, M., Tsoukiàs, A., and Vincke, P. Preference Modelling. *State Art Mult. Criteria Decis. Anal.*, pages 1–49, 2005.

Parra, D. and Brusilovsky, P. User-controllable personalization: A case study with SetFusion. *Int. J. Hum. Comput. Stud.*, 78:43–67, 2015.

Pearl, J. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Theor. Asp. Reason. about Knowl.*, volume 3, pages 121–135, 1990.

Peintner, B., Viappiani, P., and Yorke-Smith, N. Preferences in interactive systems: Technical challenges and case studies. *AI Mag.*, 29(4):13, 2009.

Pernel, D. Le corpus {«}PLUS{»}. *Notes Doc. du LIMSI*, pages 19–91, 1991.

Péry-Woodley, M.-P. *Textual clues for user modeling in an intelligent tutoring system*. {UMIST}, Centre for Computational Linguistics, 1993.

Pigozzi, G., Tsoukiàs, A., and Viappiani, P. Preferences in artificial intelligence. *Ann. Math. Artif. Intell.*, pages 1–41, 2015.

Pommeranz, A., Broekens, J., Brinkman, W. P., and Jonker, C. M. Designing interfaces for explicit preference elicitation: A user-centered investigation of preference representation and elicitation process. *User Model. User-Adapted Interact.*, 22(4-5):357–397, 2012.

Poriya, A., Patel, N., Bhagat, T., and Sharma, R. Non-Personalized Recommender Systems and User- based Collaborative Recommender Systems. *Int. J. Appl. Inf. Syst.*, 6(9):22–27, 2014.

Prince, V. *Vers une informatique cognitive dans les organisations: le rôle central du langage*. Collection Sciences Cognitives. Masson, 1996.

Prince, V. Analyse de structures de surface pour la construction d'un modèle automatique d'analyse et de production d'explications. *Interact. Cogn.*, 9(10): 107–146, 1999.

Prince, V. and Pernel, D. Querying Yellow Pages in Natural Language : A Corpus based Modelling. In *Cogn. Sci. Nat. Lang. Process.*, pages Pp.114–121., 1994.

Priss, U. and Old, L. J. Modelling lexical databases with formal concept analysis. *J. Univers. Comput. Sci.*, 10(8):1–19, 2004.

Pu, P. and Chen, L. User-involved Preference Elicitation for Product Search and Recommender Systems. *{AI} Mag.*, 29(4):93, 2009.

Pu, P., Chen, L., and Hu, R. Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Model. User-adapt. Interact.*, 22 (4-5):317–355, 2012.

Resnick, P. and Varian, H. R. Recommender Systems. *Commun. ACM*, 40(3): 56–58, 1997.

Ricci, F., Cavada, D., Mirzadeh, N., and Venturini, A. Case-based travel recommendations. *Destin. Recomm. Syst. Behav. Found. Appl.*, pages 67–93, 2006.

Roubens, M. and Vincke, P. *Preference Modelling*, volume 101. Springer Berlin Heidelberg, 1985.

Schiffer, S. Ceteris paribus laws. *Mind*, 100(397):1–17, 1991.

Schmid, H. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *New Methods Lang. Process.*, volume 12, pages 44–49. Manchester, UK, 1994.

Shoham, Y. A Semantical Approach to Nonmonotic Logics. In *Symp. Log. Comput. Sci.*, pages 275–279. Morgan Kaufmann Publishers Inc., oct 1987a.

SHOHAM, Y. Nonmonotonic Logics: Meaning and Utility. In *IJCAI*, volume 10, pages 388–393, 1987b.

SHOHAM, Y. Reasoning About Change, 1988.

SLOVIC, P. The construction of preference. *Am. Psychol.*, 50(5):364–371, 1995.

SMYTH, B. Case-Based Recommendation. *Adapt. Web*, 4321:342–376, 2007.

TAN, S. AND PEARL, J. Specification and evaluation of preferences for planning under uncertainty. Technical Report SS-94-06, Association for the Advancement of Artificial Intelligence (AAAI), 1994.

THE, A. N. AND TSOUKIÀS, A. Numerical representation of \emph{PQI} interval orders. *Discret. Appl. Math.*, 147(1):125–146, 2005.

TRABELSI, W., WILSON, N., BRIDGE, D., AND RICCI, F. Preference Dominance Reasoning for Conversational Recommender Systems: a Comparison Between a Comparative Preferences and a Sum of Weights Approach. *Int. J. Artif. Intell. Tools*, 20(04):591–616, 2011.

VAN BENTHEM, J., GIRARD, P., AND ROY, O. Everything Else Being Equal: A Modal Logic for Ceteris Paribus Preferences. *J. Philos. Log.*, 38(1):83–125, 2009.

VAN DER TORRE, L. AND WEYDERT, E. Parameters for utilitarian desires in a qualitative decision theory. *Appl. Intell.*, 14(3):285–301, may 2001.

VIAPPIANI, P. Preference Modeling and Preference Elicitation. In *First Int. Work. Decis. Mak. Recomm. Syst.*, pages 1–7, 2014.

VIAPPIANI, P., FALTINGS, B., AND PU, P. Preference-based search using example-critiquing with suggestions. *J. Artif. Intell. Res.*, pages 465–503, 2006.

VOLKOVA, S., WILSON, T., AND YAROWSKY, D. Exploring Demographic Language Variations to Improve Multilingual Sentiment Analysis in Social Media. In *EMNLP*, number October, pages 1815–1827, 2013.

VOLKOVA, S., BACHRACH, Y., ARMSTRONG, M., AND SHARMA, V. Inferring Latent User Properties from Texts Published in Social Media. In *AAAI*, pages 4296–4297, 2015.

VON NEUMANN, J. AND MORGENSTERN, O. *Theory of Games and Economic Behavior*, volume xviii. Princeton University Press, Princeton, {NJ}, {US}, 1944.

VON WRIGHT, G. H. *The Logic of Preference: An Essay*. Edinburgh University Press, 1963.

WILSON, N. Extending {CP-nets} with stronger conditional preference statements. In PETZALL, A., editor, *AAAI*, volume 4, pages 735–741. Association for the Advancement of Artificial Intelligence (AAAI), Springer, apr 2004.

Wilson, N. Efficient inference for expressive comparative preference languages. *IJCAI Int. Jt. Conf. Artif. Intell.*, pages 961–966, 2009.

Yager, R. Entropy and specificity in a mathematical theory of evidence. *Int. J. Gen. Syst.*, 219:291–310, 1983.