GOAL AND PREFERENCE IDENTIFICATION IN NATURAL LANGUAGE EXPRESSIONS

FATIMA ALABDULKAREEM

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

NOVEMBER, 2015

Abstract

Goal models allow efficient representation of stakeholder goals and alternative ways by which these can be satisfied. Preferences over goals in the goal model are then used to specify criteria for selecting alternatives that fit specific contexts, situations and strategies. Given such preferences, automated reasoning tools allow for efficient exploration of such alternatives. Nevertheless, to be amenable to such automated processing, goals and preferences need to be specified in a formal language, making automated processing inaccessible to the very bearers of goals and preferences, i.e., the stakeholders. We combine natural language processing techniques to allow specification of preferences through natural language statements. The natural language statement is first matched through regular expressions to distinguish between the preference component and the goal component. The former is then mapped to a preferential strength measure, while the latter is used to identify the relevant goal in the goal model through statistical semantic similarity techniques. The result constitutes a formal representation that can be used for alternatives analysis. In this way, stakeholders can access advanced goal reason-

ing techniques through simple natural language preference expressions, facilitating their decision making in various requirements analysis contexts. An experimental evaluation with human participants shows that the proposed system is of substantial precision and that a mapping from natural preferential verbalizations to predefined preferential strength labels is possible through sampling from crowds.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Sotirios Liaskos, for his guidance, advice, understanding, motivation, and continuous support. It would not have been possible to finish this thesis without his help and encouragement. I would also like to extend my deep appreciation to my co-supervisor, Prof. Nick Cercone, for his insights and valuable lectures in Computational Linguistics course.

I wish to thank my committee members, Prof. Marin Litoiu and Prof. Zijiang Yang for their comments and questions.

A special thanks to my family, words can't express how grateful I am to my parents for the support they provided me through my entire life. Last, but not least, I would like to express my warmest appreciation to my husband for his unlimited support.

This research has been funded by King Fahad Medical City, Saudi Arabia and the Saudi Cultural Bureau in Canada.

Part of this thesis was published in the 23rd IEEE International Requirements Engineering Conference (RE), 2015 [1].

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter One

Introduction

1.1 Goal Models and Preferences

Goal models [2, 3] have been receiving an increasing amount of attention by the research community. They have been found to be effective in modeling early phase requirements due to their usefulness in identifying and analyzing stakeholder goals, evaluating existing alternatives, supporting decision making, and understanding why the system-to-be is needed [4]. They are particularly suitable for modeling non-functional requirements [5] such as flexibility, performance and usability. Thus, in the past decade, several goal modeling languages and notations have been proposed, such as the NFR framework [6, 5], *i** [7], KAOS [2], Tropos [8, 9], and GRL [10].

A core characteristic of goal models is their ability to efficiently represent a large number of alternative ways by which stakeholder goals can be met [11, 12]. Various

contribution and constraint relationship constructs within such models allow modelers to show how low-level decisions impact the satisfaction of higher-level goals. Conversely, specified priorities over higher-level goals indicate, at the lower level, alternatives that are more preferred than others with respect to those priorities.

Consider the meeting scheduling problem, where a meeting initiator wants to organize a meeting for a group of people. The initiator will need to perform activities such as gathering everyone's constraints, putting the constraints together to decide on a suitable time slot, booking a room and inviting the participants. There may be several different ways to perform each of these activities. For example, to achieve the goal *"Constraints Gathered"*, the organizer can either follow a *"Request Based"* approach, where s/he sends an invitation for constraints and waits for responses, or *"Consult On-Line Calendars"* in which s/he simply assumes that everyone has their on-line calendars up-to-date and simply retrieves and consults those to decide a slot. Each of these possibilities contributes to different higher-level objectives of the meeting organizer or other stakeholders. If the stakeholders prefer to *"Reduce Labor"*, then they need to choose the *"Consult On-Line Calendars"* path. However, if the stakeholders prefer to *"Maximize Attendance"*, then they probably need to choose *"Request Based", "Call Everybody"* path which will contributes positively to the *"Maximize Attendance"* at the cost of contributing negatively to the *"Reduce Labor"*.

From the above example we can see how choosing between alternative low level

goals affects the contribution to the high-level goals. Formal preference specification techniques have been proposed for capturing such priorities in a way that allows automated search and identification of suitable alternatives [13, 14]. Continuing with the same example, if the stakeholders prefer to achieve *"Reduce Labor"* goal, this desire can be be translated into formal preference specification as below:

$$\Box(valS(ReduceLabor) \geq 0.8) \tag{1.1}$$

This process of exploring alternatives that match user preferences can be useful for supporting decisions during early requirements engineering – e.g., deciding which socio-technical design to pursue [12] – or later in the lifecycle when a system requires reconfiguration to meet changing needs, situations and contexts in a requirements-driven way [15, 16]. In both cases, automated reasoning can efficiently identify solutions that best match individual problem specifications.

1.2 Specifying Preferences: Challenges

Although formal preference specification appears to be a solution for dealing with large spaces of alternatives at a high-level, it has three barriers that may make it difficult for non-technical stakeholders to use. First, preference specifications are formal, meaning

that they have to adhere to specific syntax and to use of terms that are embedded in the preference language. Second, because preference specifications are using terms taken out of the goal model, they assume access to and knowledge of that goal model and the exact phrasing of goals. Third, preference specifications require an arbitrary formalization of preferential strength (how strongly something is preferred or not preferred) into a label or number.

For example, in a formal preference specification as in Liaskos et al. [14] to express the goal *"Reduce Labor"* from a meeting scheduler domain with a high preferential strength, the user needs to use Optional Condition Formulae (OCFs) and Linear Temporal Logic (LTL) as below:

$$\Box(valS(ReduceLabor) \geq 0.8) \tag{1.2}$$

The LTL symbol $\Box$ refers to the temporal operator "always", while *valS* indicates that the satisfaction value of the associated goal of *"Reduce Labor"* should be with a high value (0.8).

In the above preference specification we need to be aware of the exact syntax, i.e., where to put the parentheses, how to spell valS, use the symbol for always, form the preferential strength label, and also know the exact phrasing for the goal as it was mentioned in the goal model, i.e., "Reduce Labor" rather than "Reduce Work" or "Save effort".

Formal languages such as the above do not seem to facilitate the development of usable goal-oriented decision exploration tools that can be used by non-technical users. In many potential applications of preference specification such as software customization and adaptation, users need to be able to specify their preferences without the need of an expert to translate their preference expressions manually in a machine-readable form. We alleviate this by designing an NLI that allows novice users to express goals and preferences with natural language they use in their daily communication.

1.3 Specifying Preferences using Natural Language

We propose to identify stakeholder preferences through processing natural language expressions generated by the stakeholders themselves. We begin by assuming the existence of a goal model that describes a domain of intention for a particular stakeholder, such as a goal decomposition model for achieving the goal *Schedule Meeting*. The model is constructed by experts and its details need not be accessible by the stakeholder; a meeting organizer in our example. In a meeting scheduling context the organizer may wish to identify solutions (i.e., ways to satisfy top level goals) in which certain objectives are emphasized, i.e., certain goals are more important than others. In our proposal, instead of exploring the visual goal model and formulating formal preference statements, our stakeholder simply specifies her interests in natural language, in ways such as *"it is important to schedule quickly"*, *"it is OK to use on-line calendars"* or *"sending reminders is not*

necessary", which do not necessarily adhere to a specific syntax or vocabulary. Subsequently, our proposed system uses common regular expressions to split the goal part of such statements, which refers to the goal that the stakeholder wants to achieve, from the preference part, which refers to how strongly she wants to achieve the goal. The former is matched with the semantically closest goal in the goal model, using knowledge-based-supported statistical semantic similarity techniques [17], while the latter is matched with a numeric or qualitative label through reference to a corpus of labeled natural preference expressions. The two results combined constitute a preference statement that can be used for formal reasoning. In this way, stakeholders can reason about alternatives without the need to access the underlying goal model.

This possibility of accessing goal models through natural language may find many applications in practice. For example there is work on evaluating goal models using its visual representation, such as in Horkoff et al. [18] where the user needs to navigate the goal model to locate goals of interest. Our approach can take such proposals one step further and allow locating those goals and specifying how important they are without manipulating the goal model. This enables the user to indirectly specify alternatives by just specifying a group of preferences and how important they are. Another application is configuration. It has been proposed [16] that goal models can be effective in mediating between users and several aspects of software configuration. An example of this is a complex computing platform, whose administrators need to configure it to address

multiple workload, security and failure situations through specifying what the goals of a reconfiguration are: e.g., increase throughput, increase response time, secure availability, etc. In systems configured like this, through using our proposed method, the users can change the configuration by specifying the new preferences in natural language, lifting thereby the need for them to comprehend and manage the complexity of a large number of configuration variables.

In the empirical evaluation we conducted, we used rephrasing of pre-constructed preferences as a proxy for natural specification of preferences. In particular, experimental participants are introduced to a domain and then asked to rephrase goals and preferences that are associated to that domain. We then test whether the rephrasing are understood by our system. Among other things, we observe that goals and preferences are recognized with notable precision and that the use of crowd-sourced corpora of labeled natural preferential expressions can actually be effective for identifying preferential strength.

1.4 Contributions

Our contribution lies in three potential areas:

- From a goal analysis standpoint, we introduce a natural language technique that could increase the accessibility of preference-based and – after properly adapted – potentially of other kinds of goal reasoning toolkits.

- From a requirements prioritization viewpoint, we offer a way to elicit strengths of goal preferences using natural language, which may be added to the range of tools that have been proposed in the area.

- Finally, from a variability analysis and software customization point of view, we extend earlier proposals for preference-based software customization with an interface that could allow non-technical users perform system customizations through natural language.

1.5 Thesis Organization

This thesis is organized as follows. In Chapter 2 we offer an introduction to goal models and preferences, after that we explain the natural language processing techniques that we adopt in our system, then we review the existing literature which have similar approaches. In Chapter 3 we have an overview of our solution, then we describe in details our designed system including regular expression, preferential strength identification, building preference repository, semantic similarity, negation identification, and post processor. In Chapter 4 we present the design and results of our evaluation then we discuss the threats to validity. Then, we conclude in Chapter 5.

Chapter Two

Background

2.1 Goal Modeling and Variability Analysis

Requirements Engineering uses a variety of approaches to elicit, analyze, specify, validate, and manage stakeholder requirements. Many of these approaches focus on the late phase of requirements analysis, i.e., producing a complete, precise, and consistent requirements document by focusing on what the desired system is supposed to do and how to do it. However, an equally interesting problem is that of exploring system solutions and alternatives, analyzing stakeholder interests, and identifying problems. These activities are characteristic of the *early phase requirements analysis*, i.e., the phase prior to the initial requirements formulation and when the features of the system have not yet been identified [2, 3]. To assist the early requirements engineering effort, goal-oriented requirements engineering (GORE) approaches have been used to identify, analyze, de-

scribe, document, and modify goals. The central activity in GORE is goal modeling, i.e., describing how goals can elicit, communicate, evaluate alternatives, and describe high level requirements. Below we describe several goal modeling frameworks such as NFR framework, *i**, and GRL.

2.1.1 NFR Framework

A desired system is defined by its functional requirements as well as by non-functional requirements (NFRs) such as usability, integrity, and performance. However, most of the attention paid in software engineering in the past has been to the functional requirements that the system-to-be needs to perform, although the functional requirement cannot develop a high quality system on its own; it needs to be combined with non-functional requirements.

In recent years, researchers have been investigating methods for identifying NFRs. One of the proposed solutions was the NFR framework [6, 5].

The NFR framework emphasizes representing, organizing, and analyzing non-functional requirements by using the concept of *softgoal* to represent them. Softgoals are goals for which there is no clear cut criterion to decide whether they are satisfied or not.

In the NFR framework, a softgoal is considered satisficed [19] (i.e., attaining satisfaction defined as falling within acceptable limits rather than as an absolute) if there is a sufficient positive evidence and little negative evidence against it. There are three types of

softgoals in this framework: *NFR softgoals* that represent non-functional requirements; *operationalizing softgoals* that express possible solutions for satisficing NFR softgoals; and *claim softgoals* that clarify the rationale for a softgoal.

To analyze softgoals, they need to be in relation with each other. In order to achieve this, two types of interdependencies are used. The first type is the AND/OR decomposition, while the second type is used to describe looser relationships where a softgoal prevents or contributes to the fulfillment of others. This looser relation is illustrated by the following labels: *Make*, *Help*, *Break*, and *Hurt*. The *Make* value indicates that a softgoal is fully satisficing another softgoal; the *Help* value indicates that the softgoal is partially satisficing another softgoal; the *Break* value indicates that a softgoal is fully denying another softgoal; and the *Hurt* value indicates that a softgoal is partially denying another softgoal [20].

To keep track of all types of softgoals and their interdependencies, a Softgoal Interdependency Graph (SIG) is created and maintained. In the SIG, softgoals are represented as nodes; their interdependencies are represented by links.

In addition, SIGs are used to form NFR catalogues that contain the knowledge base for non-functional requirements [21]. There are three types of catalogues: *NFR type catalogues* represent knowledge about a particular type of NFRs; *method catalogues* gather information about techniques to help refine softgoals; and the *correlation catalogues* show the implicit interdependencies between softgoals. These catalogues are useful in

the software development phase, as well as in the reengineering work during the software life cycle.

2.1.2 *i** Framework

*i** is an agent-oriented goal modeling framework. It provides modeling and reasoning about organizations by showing the dependency relationship between actors. The name *i** refers to the notion "distributed intentionality".

The *i** framework models early phase requirements by representing the stakeholders' objectives and the relationships between them, laying out the current organizational situation and why the system-to-be is needed. *i** integrates concepts from the NFR framework, such as softgoals, contribution links, and dependencies between actors.

The main concept in *i** is intentional actors. These entities have attributes such as goals, abilities, beliefs, and commitments. The actors rely on each other for help in achieving difficult goals or tasks that an individual actor cannot achieve alone.

This framework consists of two modeling components, the Strategic Dependency and the Strategic Rationale, which can be represented in the conceptual modeling language Telos [22, 3, 7].

2.1.2.1 The Strategic Dependency (SD)

The strategic dependency model expresses the intentional dependency relationships between

a range of actors in the organization. This model consists of a set of nodes and the links between them. Nodes represent actors, while links indicate that an actor depends on other actor.

Actors rely on each other to achieve goals, perform tasks, and produce resources. Thus, an actor can achieve more by depending on other actors. An actor can be classified as an *agent*, which is an instance of human, machine, or software; *role*, which is the abstract characterization of actor; and *position*, which is a set of roles played by one agent. The dependencies between actors have four types based on the subject of the dependency: goals, softgoals, tasks, and resources [7].

Actors are intentional and strategic, since they have wishes and concerns about opportunities.

2.1.2.2 The Strategic Rationale (SR)

The strategic rationale model expresses how to address stockholder interests and concerns by the use of alternative configurations. The SR model analyzes the internal process of each actor in detail, while the SD model provides only the external relationships between actors.

SR uses *Task-decomposition* links and *Means-ends* links as its main relationships. These links are used to model AND and OR decompositions, respectively. *Means-end* links provide information about why an actor would perform a task, goal, softgoal, or

resource. They give deeper understanding and show possible alternatives that give the actor the ability to choose between them. The means is usually a task, whereas the end can be a goal, task, resource, or softgoal. Means-end links have several types: Goal-Task, Resource-Task, Task-Task, Softgoal-Task, Softgoal-Softgoal, and Goal-Goal. The impact of an element on a softgoal is expressed by contribution links which have similar meaning to the ones in the NFR framework. However, the contribution links in *i** have different labels that are ("++" and "+") which indicates two levels of positive and ("--" and "-") which indicated two levels of negative [7].

Task-decomposition links connect a task to its decomposed sub-components, which can be tasks, goals, softgoals, or resources. Each decomposition link can be *open*, *committed*, or *critical*. *Open* means that the routine could be affected but not necessarily fail, while *committed* means that the agent believes that the failure of the element will result in the failure of the routine; *critical* means that the agent believes the element is necessary for success.

2.1.3 The GRL Framework

Goal-oriented Requirement Language (GRL) is a language for supporting goal- and agent-oriented modeling and reasoning with regard to requirements, especially non-functional requirements [6]. GRL is an elaborated version of *i**; it also integrates concepts from the NFR framework to model non-functional requirements.

GRL uses many concepts of *i**, such as actors, intentional elements, dependencies, contributions, and decompositions. It also uses the same graphical syntax. However, there are some differences between the two frameworks. The first is that *i** has different types of actors—roles, agents, and positions—whereas GRL has only one type of actor. The second difference is that in GRL intentional elements can be linked to each other in various ways, while *i** is more restrictive, e.g., contribution links can only have a softgoal as an end. Also, GRL has one model with multiple views and does not distinguish between Strategic Dependency (SD) and Strategic Rationale (SR) models, as is the case in *i**. Moreover, GRL has unique concepts that are not used in *i** such as strategies, metadata, URN links, and an importance factor, which—when incorporated with the satisfaction level— help to measure the satisfaction of an actor [23].

GRL actors aim to achieve goals by executing actions. The actor can be associated with the goal graph, i.e., a connected graph of intentional elements, by circling the graph with a dashed line. The intentional elements in GRL consist of five elements: goals, softgoals, tasks, resources, and beliefs. The *goals* and *softgoals* are similar to those in the earlier frameworks. The *tasks* are solutions to goals or softgoals, while the *resources* are often required to achieve a goal, softgoal, or task; the *beliefs* are facts that are important to the stakeholder.

To connect elements in the goal model, intentional relations are used. These relations are contribution, decomposition, dependency, correlation, and means-end. *Contribution*

links are used to describe how the intentional elements contribute to each other. These links have qualitative or quantitative contributions. The qualitative contribution links types include: *Make*, *Help*, *Break*, *Hurt*, *Some Positive*, *Unknown*, and *Some Negative*; the first four contributions are similar to the contributions in NFR framework. The quantitative contributions are integer values between -100 and 100. *Decomposition* links define the sub-components of an element. *Dependency* links model the relationship between actors. *Correlation* links describe side effects of one element on other. *Means-end* links describe how goals are achieved. Each task connected to means-end link describes an alternative for achieving that goal [10].

GRL elements and relations are intentional because they are used to answer questions about motivations and rationales. There are major benefits to GRL over other frameworks, including its support of qualitative and quantitative satisfaction, the integration of scenario notation, and the clear separation of elements from their graphical representation that allows multiple views of the same goal model [24].

GRL becomes a standard part of the User Requirements Notion (URN), which is a recommended standard of the International Telecommunications Union [24]. URN combines the *Goal-oriented Requirement Language* (GRL) for its ability to model goal-oriented concepts, with *Use Case Map* (UCM) notation for modeling scenario concepts into one language.

2.2 Goal Modeling for Alternative's Analysis

The work we describe in this thesis does not necessitate commitment to any one of the above frameworks. Thus, although we generally comply to *i** the ideas we present are applicable to all the above dialects. Specifically, the types of models we consider here consist of AND/OR decomposition trees, that show alternative ways by which high-level stakeholder goals can be fulfilled, as well as softgoals that are used to capture the impact of alternatives to higher level objectives. One such model can be seen in Figure 2.1. In the model, hard-goals (the ovals in the figure) are decomposed into other hard-goals or tasks (hexagonal elements) through AND- and OR-decompositions, resulting in trees that describe many alternative ways by which the root goal can be satisfied. Furthermore, softgoals (cloud-shaped elements in the figure), which as we saw earlier, are goals of a less precise definition, receive positive and negative contributions from other goals, making the satisfaction of the former depend on the choice of the latter. Thus, by identifying certain goals as more important than others, the goal model implies that certain alternatives are also more suitable than others. But how can we more precisely specify such preferences between goals? We discuss this in the next section.

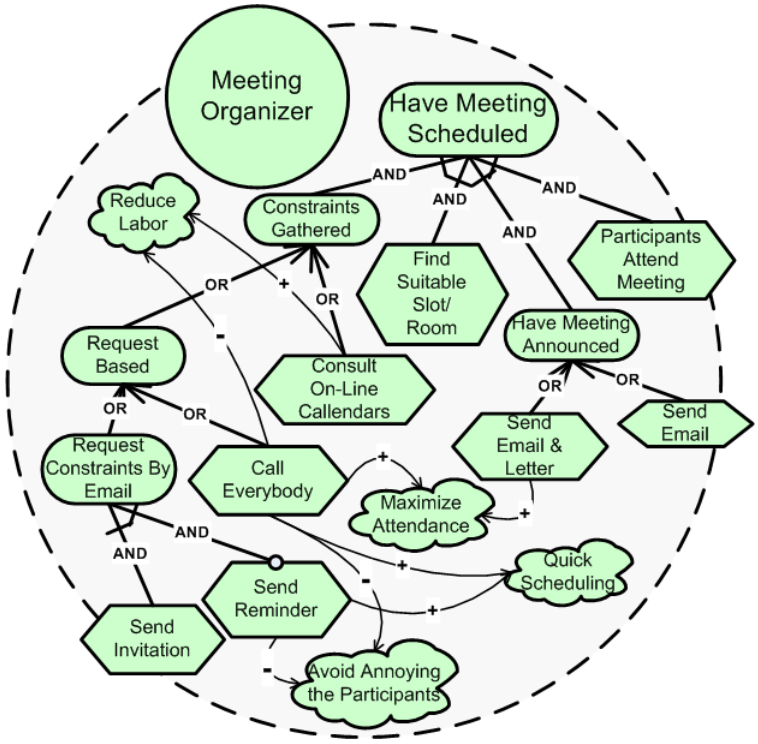


Figure 2.1: A Goal Model

2.3 Specifying Preferences

As we saw above, selecting the appropriate alternative is a result of specifying preference between higher-level goals and qualities. To formalize such a specification and automate the search for suitable alternatives, preference and priority specification and analysis techniques has been proposed [13, 14]. In that work, the emphasis of a stakeholder to a specific goal over others is expressed by creating preference statements and then combining them in priority relations. The latter are, in turn, used by automated reasoning tools to identify appropriate alternatives. While several versions of such formulations

have been proposed as we explained earlier in 1.2, the essence of such specification is that (a) some goals are picked out by the stakeholder as worth mentioning, (b) the stakeholder expresses some degree of desirability for each of the picked goals.

Returning to our example, in a specific meeting scheduling scenario, the stakeholder may express the statements *"holding it as quickly as possible is crucial"* and *"it is OK if we do not use on-line calendars"*. These expressions alone need processing before they can be useful for automatic reasoning. Thus, preference specification requires us to identify the goals in the goal model to which the stakeholder might refer. These are *Quick Scheduling* and *Consult On-Line Calendars*, in our case. Then the translation of the expressions of importance *"[...] is crucial"* and *"it is OK if [...]"* into machine recognizable labels (quantitative or qualitative) needs to be performed. If we assume the former to be 0.8 and the latter 0.2, then the complete preference would look like: {`Quick Scheduling [0.8], ¬ Consult On-Line Calendars [0.2]`}. An expression like this can be adapted for use by an automated reasoner (e.g., a planner-based [13, 14] or a SAT solver-based [25]) for identifying alternatives. But the formulation was done manually, probably by an analyst/expert, who needs to be aware of the goal model and the mapping from expression of preferential strengths to labels.

In this thesis, we explore how we can allow machine, rather than manual, translation of natural preference expressions into formal statements. To achieve that, we use a combination of techniques, which we introduce below.

2.4 Semantic Similarity in Natural Language

Our proposed natural preference expression processing system is based on (a) identifying, distinguishing, and splitting the goal and preference components within the natural language statement entered by the user, (b) identifying the goal in the goal model that the statement most likely refers to, (c) associating the expression with a predefined preferential strength label. In our proposal, part (a) is accomplished through the use of *regular expressions*, part (b) through *statistical semantic similarity* analysis and part (c) by looking up a preference key-phrase repository defined through examining sample expression cases. In the rest of the chapter we offer an overview of the technologies we adopt to perform the above tasks.

2.4.1 Regular Expressions

Regular expressions are search patterns specified in the form of specially constructed sequences of characters. They are remarkably common in many applications in computer science which involve e.g., checking if an input matches a text pattern, splitting a text, etc., and have been shown to be useful in areas such as information retrieval [26] and web semantics [27].

The regular expressions we consider here consist of regular characters, which have a literal meaning as well as meta-characters which have a special meaning. These meta-

characters could be used individually or combined together to form a search pattern. In our application we adopt Python's regular expression module and use many of the available meta-characters, particularly ? . * + | ( ) \W \s. An example of regular expression patterns can be found in Table 2.1.

Table 2.1: Regex Patterns with Examples

Pattern	**Description**	**Example**	**Regex**
*	Matches 0 or more repetitions of any characters	tree and trends	tre*
?	Matches 0 or 1 repetitions	color and colour	colou?r
|	Matches between either sets	a or b	a|b
\s	Matches any white space	the sentence is matched with regex	[a-z\s]+
\W	Matches any non-alphabetic and non-numeric character	doesn't and doesnt	doesn[\W]?t
^	Matches beginning of line	Match "tree" at the beginning of line	^tree
$	Matches end of line	Match "tree" at the end of line	tree$

2.4.2 Matching through Semantic Similarity

Semantic similarity is used in natural language processing to measure the similarity of meaning between words and phrases. There are numerous methods for performing semantic similarity [28]. One class of such techniques uses the "bag of words" [29], the modeling of texts through vectors containing its words. In what is called the co-occurrence method, the cosine similarity of the vectors is calculated to get the similarity between the texts. A more refined method is the descriptive features-based method that represents each word in a sentence by a set of predefined features [30, 31] such as word

property and word frequency. The text is then represented by a vector consisting of these features values. Corpus-based techniques, on the other hand, such as Latent Semantic Analysis (LSA) [32] and Hyperspace Analogues to Language (HAL) [33] rely on information in large *corpora* (i.e., large collections of "authoritative" texts) in order to establish similarity.

We adopt the technique used in the UMBC Semantic Textual Similarity Service (UMBC STS) developed by Han et al. [17], which combines LSA with WordNet [34] in order to account for multiple meanings of words. The UMBC STS adopts LSA by constructing a word-by-word co-occurrence matrix based on the analysis of a large corpus (Stanford WebBase project [35]). They used the Stanford WebBase collection from February 2007 that contains 100 million web pages from more than 50,000 websites. The total number of words after processing for duplication, non-English text, and unknown characters were three billion [36].

The co-occurrence matrix is constructed by sliding a window of $\pm$N words, one word each time over the entire corpus and increasing the frequency in the appropriate cell when two words co-occur in the window. Based on the hypothesis that words occurring in the same contexts tend to have similar meanings [37], e.g., "car" and "driver" or "wife" and "marry", high-values in the co-occurrence matrix imply relatedness. The window size can have a small or large parameter depending on the purpose for the model itself. Large window size captures word relation within the same general topic in the document, while

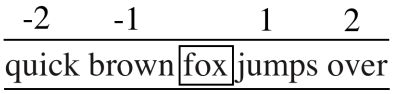


Figure 2.2: Window Size of ± 2

small window size can captures the grammatical relation between words. An example of window with a size of ± 2 is illustrated in Figure 2.2.

In the UMBC STS service there are two models generated according to the word/term co-occurrences in a moving window. The first model called the *concept model*, which has a window size of ± 1 and can detect semantic similarity within the same POS (e.g., "car" and "vehicle"). The other model called the *relation model*, which has a window size of ± 4 and can detect semantic similarity between words of different POS (e.g., "eat" and "food").

Furthermore, as co-occurrence itself seems to be insensitive to alternative senses (meanings) of words such as "the bank is constructed from red brick" and "I withdrew money from the bank", the UMBC engine utilizes WordNet [34], a large lexical database of English, to draw additional similarity evidence through relations identified there such as synsets (synonym sets) or hypernyms (general-specific relations). The UMBC STS service requires minimum similarity between two words in the LSA model of 0.1 to extract relations from WordNet. Since WordNet lists several senses for each word, UMBC STS service defines "significant senses" to be used in their system. They consider a sense to be significant if it matches one of the following: (a) the word is the first sense of the

other word; (b) the word frequency count in WordNet is not less than five; (c) the word sense number in WordNet is less than eight and its word form is the first in its synset word form list.

Given two input sentences the UMBC engine uses metrics based on the resulting similarity measures to conclude whether the sentences are semantically similar or not. The semantic text similarity is indicated by a numeric value in the interval [0,1], where 0 signifies that the sentences are totally dissimilar and 1 that they are identical. Thus, the sentence *"Invitees Join Meeting"* is similar to *"Participants Attend Meeting"* by 0.56, and similar to *"Find Suitable Room"* by 0.24.

2.5 NLP-based Goal and Preference Identification: Existing Approaches

Natural language processing techniques have been widely used in requirements engineering for a variety of purposes. For example, Cleland-Huang et al. [38, 39] use a supervised classification algorithm to identify non-functional requirements within structured and unstructured texts while Casamayor et al. [40] use a semi-supervised learning techniques to identify non-functional requirements by training the classifier with a small number of manually identified requirements. Breaux and Antón [41] propose generating goal models from natural language policies by achieving two steps goal mining and semantic parameterization. On the other hand, Weber-Jahnke and Onabajo [42] use semantic annotation ontology to analyze natural language confidentiality requirements.

Elsewhere, Yang et al. [43] use natural language techniques to detect uncertainty and speculative sentences in stakeholders natural language requirements.

Furthermore, the problem of interacting with goal models has received some attention from the requirements community as well. Horkoff and Yu, particularly, have proposed an interactive algorithm for evaluating goal satisfaction within goal models [44] and have performed studies exploring various visual aspects of goal modeling [45].

Goal detection has been a research topic in other communities as well. Y. He, for example, [46] uses Tree-Augmented Naive Bayes networks (TANs) to detect goals from natural language expressions and evaluate them on two different corpora. This evaluation resulted in a higher accuracy rate compared to an earlier one that used Naive Bayes [47]. Casagrande et al. [48] use NLP and data mining techniques to extract goals from research abstractions and use them to create a taxonomy. Kröll et al. [49] on the other hand propose a system to automatically annotate text with human intentions using indicative actions as a proxy for inferring such intentions.

Natural language expression of preferences has also been studied. Using evidence from an exploratory study involving collection of examples of participant expressed preferences, Nunes et al. [50] have proposed a meta-model for natural preference formulation which they evaluated in terms of its usefulness for actual preference specification. The meta-model contains different preference types such as goals, constrains, and expressive speech act. However, the ontology meta-model did not support adjectives or subjective

specified values. In other work, they extend the meta-model to support decisions [51].

In software engineering, natural language and information retrieval techniques have been used in requirement traceability. For example, Antoniol et al. [52] apply a probabilistic and a vector space models to recover traceability links between source code and natural language documents while Gibiec et al. [53] propose an automated technique by developing a web-mining method that modifies the terms found in the trace query with new ones.

Moreover, natural language techniques have been used to detect duplicate products on the Web. Köpcke et al. [54] propose the use of machine learning algorithm and regular expressions to detect duplicates within product titles by extracting product codes. de Bakker et al. [55] use the Title Model Words Method (TMWM) [56], i.e., detects model words from product names and compares them to find duplicates, and extend it on product attributes.

In information retrieval, topic identification could be considered similar to goal detection. Lagus and Kuusisto [57], presented a method to identify topics of dialogue using neural network method, in particular Self-Organizing Map (SOM) [58, 59]. Clifton et al. [60] presented TopCat (Topic Categories) technique that identifies topics in documents by applying data mining and natural language techniques while Damashek [61] identifies documents by topics or languages using n-gram method. Bellegarda and Silverman [62] use Latent Semantic Analysis (LSA) for semantic inference. Many of information

retrieval methods could be applied in goal detection.

Finally, the use of natural language techniques to customize preferences in configurable software systems, has also been proposed [63]. The user can specify the desired preference in natural language and through a combination of techniques, including WordNet and a fast TF-IDF (Term Frequency-Inverse Document Frequency) algorithm to measure similarity.

Our work complements and extends existing efforts in a number of important ways. Firstly, we focus specifically on goal models and reasoning therewith by utilizing natural language processing for the task of evaluating goal models, which does not seem to have been attempted in the GORE literature (e.g., Horkoff and Yu above [44, 45]). Secondly our system concentrates on the detection of goals and preferences at the same time, through splitting natural language expressions into the corresponding parts. We could not find literature that attempts exactly that, although as we saw, there is work on detecting goals and expressing preferences [46, 47, 48, 50]. Finally, we propose a way to crowdsource the mapping from preference statements to quantitative and qualitative labels, that differs from existing work (e.g., Nunes above [50]) in natural specification of preference versus a semi-structured specification approach and attempts a distinct corpus-based approach for assessing preferential strength.

Chapter Three

Solution

3.1 Overview

As we saw in the previous chapter preferences are useful for allowing stakeholders to prioritize between goals that influence the distinction between alternative solutions. However, earlier techniques introduced to specify preferences were formal and, as such, difficult for novice users to use and get the desired benefit.

In this chapter, we describe our Natural Language Processing based solution to this problem, that allows users to phrase their goals using the natural language of daily communication. Our system is based on splitting the goal from the preference component of natural language expressions using regular expressions. The goal component is then used to identify the goal in the goal model using semantic similarity techniques, while the preference component is translated into a quantitative or qualitative label through

reference to a crowd generated repository of labeled preferences. The result is a formal preference specification such as the ones that have been proposed in the literature [64].

In the following sections we start by providing an overview of the architecture of our system. Then we look in more details into each component and describe how they work together to produce the end result.

3.2 Architecture

The architecture of the proposed system can be seen in Figure 3.1. The natural expression given by the user is first processed by the *Regex* module on the left of the figure. This module distinguishes between the goal and the preference components of the natural language expression that the user provides – we call this process *splitting*. The outputs of the Regex module are, thus, two: the *goal part* and the *preference part* of the original expression. The goal part is given as input to the Semantic Similarity engine implemented through a call to the *UMBC STS Service*. The list of goals, extracted from the domain goal model is also given as input to the engine. The engine identifies the goal in the goals list that has the highest similarity score with the goal part of the original expression. Meanwhile the preference part is passed to a *Preference Identification* module in order to match it with a strength label. At the same time, the goal part is passed to a *Negation Identification* module. Eventually, the matched goal from the goals list, the preference strength level as well as the extracted negation information are given to a

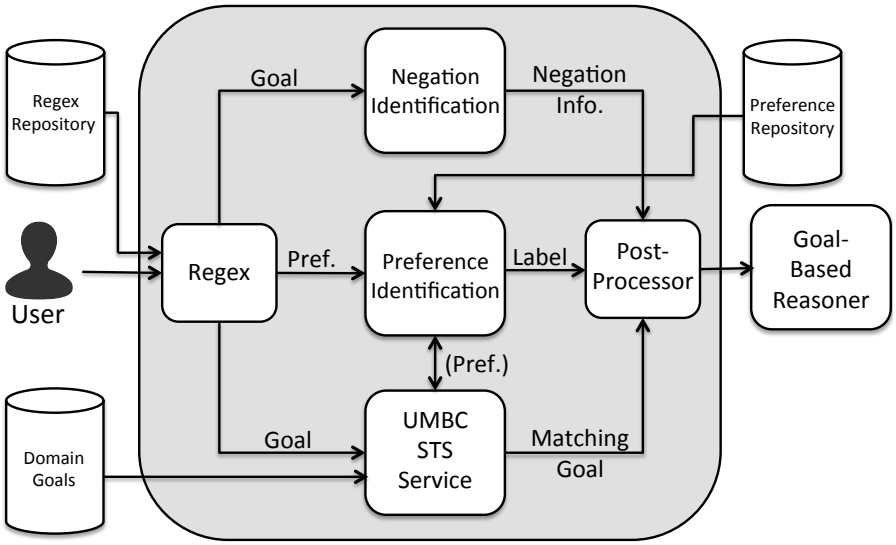


Figure 3.1: System Architecture

post-processor, which constructs a formal preference. We describe these in more details below.

3.3 Regex

The Regex module splits the natural expressions through looking-up and applying a repository of regular expressions, the *Regex repository*. The regular expressions are constructed manually through the study of example natural language expressions of preference. Their structure depends on the order by which the goal and its preference occur in natural expressions. We found that many natural expressions of preference could be classified into three categories: preference expression followed by a goal (*"it*

is $\{$*important*$\}_{Pref.}$ *to* $\{$*drive carefully*$\}_{Goal}$*"*), a goal followed by a preference expression (*"*$\{$*driving carefully*$\}_{Goal}$ *is of* $\{$*high importance*$\}_{Pref.}$*"*), or just a goal without a preference (*"drive carefully"*). An example of the splitting process is presented in Table 3.1.

Given the Regex repository, the Regex module sends the natural expression provided by the user to a pattern matching function. The function will search through the repository to find a suitable rule. The search is sequential and once a match is found the function will stop checking the rest of the regular expressions. As such, the regular expressions in the repository are ordered from specific to general.

After finding the corresponding regex for the sentence, the function will match whatever between the placeholders (.*?) and (.*) with the equivalent words in the sentence.

Furthermore, the system will start by checking rules that contain a preference fol-

Table 3.1: Splitting Sentences with Regex

Sentence	**Preference**	**Goal**	**Regex**
Using on-line calendars is OK	OK	using on-line calendars	(.*) is (.*)
The importance of sending emails is high	high	sending emails	the importance of (.*) is (.*)
I'm interested in having the meeting scheduled quickly	interested	having the meeting scheduled quickly	i[\W]?(am|m) (.*?) in (.*)
It is of high importance to maximize the number of the attendees	high importance	maximize the number of the attendees	is of (.*?) to (.*)
Announcing the meeting should be done quickly	should be done quickly	announcing the meeting	(.*) should be done [a-z\s]*
Ignore entirely sending reminders	ignore entirely	sending reminders	(disregard|ignore entirely) (.*)

lowed by a goal. If no match is found, the system will move to the next category of rules where the goal is followed by the preference. If no match is found there either, the system will generate a message indicating that the preference was not found, i.e., the natural language statement entered by the user has no preference and it potentially contains only a goal.

3.4 Preferential Strength Identification

Once the preference part of the user expression is identified in Regex, its *preferential strength label* needs to be identified, i.e., the degree by which the goal part of the user expression is actually desired. As we demonstrate in our evaluation (Chapter 4), various quantitative or qualitative scales are possible for preferential strength. For the purpose of our presentation here we focus on the discrete scale {0%, 25%, 50%, 75%, 100%} to label preferential strength. Moreover, a *preference repository* holds a set of predefined phrases that express preference; for clarity we call these *preference key-phrases*. Each of those preference key-phrases is associated with a preferential strength label. Examples of preference key-phrases that emerged in our repository shown in Table 3.2.

Given an identified preference part of a natural expression, the module will try to match, in an exact fashion, the preference part with a preference key-phrase in the preference repository. Once a key-phrase is matched, then the strength label associated with it is adopted as the preferential strength label implied in the original natural expression.

Table 3.2: Preference Key-Phrases with Labels

Preference Key-Phrase	**Label**
absolutely required	100%
vital	100%
your second priority	75%
pretty important	75%
medium priority	50%
somewhat important	50%
low importance	25%
not very important	25%
we rather don't need	0%
least important	0%

Nevertheless, the preference part could be a phrase that was not found in the preference repository. In such cases, we kept all the preference key-phrases under each label in a separate files, then we execute the UMBC STS service between the unknown preference part and the set of all key-phrases under each of the five files. The label of the file which yields the highest similarity is adopted as the associated label. For example assume *"not an important aspect"* is a preference part not included in the preference repository. We evaluate its semantic similarity against all key-phrases under each of the files associated with the labels 100%, 75%, etc. (five tests) and find that it is semantically similar with the key-phrases under 0%.

3.5 Building Preference Repositories from Crowd Data

Preference repositories can be developed through crowdsourcing. The process is based on building a corpus of labeled examples of natural preferential expressions. To achieve this, we provide to a number of participants goals and a set of predefined preferential strength labels. We then ask them to produce for each given goal a suitable natural language expression of preference that matches each strength label. Thus given a goal e.g., "Schedule a Meeting" a crowd of participants is asked to naturally write examples of expressions of preferences on that goal with strengths 100%, 75%, etc. Each of those expressions is then passed to Regex to identify the preference part/phrase. The result is a collection of preference phrases, each associated with a participant-chosen strength label.

This collection is used as a corpus for classifying newly inputted preference expressions. Specifically, each phrase in the corpus is measured with respect to the frequency in which it occurs with different preferential strength labels. For example, assume that, of all the combined answers, the participants have provided twenty (20) natural language expressions with preference part being *"very important"* (e.g., from *"it is very important to schedule a meeting"* or *"scheduling a meeting is very important"*). However some of those expressions, say 12, were given by the participants under the label 100% and 8 under 75%, i.e., phrase *"very important"* occurs in the corpus 12 and 8 times with

each label, respectively. The numbers 12 and 8 are the *support* of each phrase-to-label association.

To allow representation in the preference repository of the support values calculated from the corpus, each key-phrase in the repository is associated with a label set rather than a single label we described earlier. Each element of the label set is a tuple <*Label,Support*>, where the *Label* represents the preferential strength label, and *Support* represents the number of preference phrases in the corpus that identify with the key-phrase in question and are associated with *Label*. When a new natural expression is entered, the system identifies a preferential strength to it as follows. First, the associated preference key-phrase is identified. Then in the corresponding label set, the label associated with the highest support is chosen to be the preferential strength of the original expression. Back to our example, for key-phrase *"very important"*, label 100% has support 12 and label 75% has support 8. Thus a phrase such as *"scheduling a meeting is very important"* is assigned preferential strength label 100%. A sample of the preference key-phrases and their associated labels and support numbers can be found in Appendix A.

Alternatively, one can interpret the labels X, Y as samples from a continuous scale and produce a weighted average as shown in Equation 3.1. In our application and evaluation we followed the simple majority rule.

$$X \cdot (12/20) + Y \cdot (8/20) = Z \tag{3.1}$$

3.6 Semantic Similarity

While the preference part that results from the splitting is passed to the above strength label identification process, the goal part is passed to the semantic similarity engine, the UMBC STS service, for identifying the goal in the goal model that is more strongly related to. The engine accepts pairs of phrases as inputs, and produces a similarity score using techniques we described in the previous chapter. For our purposes we compare the goal part that comes out of the Regex with every goal of the goal model and simply identify the goal that has the highest similarity score. Note that the UMBC STS service offers two models according to the word/term co-occurrences. In our system, we used the *relation model*, i.e., one that results from considering a larger corpus processing window (see Chapter 2), to detect the semantic similarity between different parts of speech (POS), which is the case for the sentences entered by the users.

3.7 Negation Identification

A challenging part in processing the natural language expression in our case is the identification of negations in the goal part of the expression. In the presence of negations, the

semantic similarity techniques that we consider here will identify, for example, expressions such as *"it is fine to have little attendance"* with the goal "Maximize Attendance", unaware of the fact that less attendance contradicts maximizing attendance.

To tackle this negation issue we populate, wherever applicable, the softgoals of the list of goals with softgoals of the opposite meaning; we call these newly introduced goals, *shadow goals*. Thus for each goal that contains words such as "reduce", "prevent", "restrict", "limit", etc. we introduce a new goal by replacing these terms with antonyms, such as "increase", "allow", "encourage", "assist", etc., respectively. The process is manual as the exact choice of antonym depends on the context; multiple antonyms of the original goal are possible. In this way, if the user refers to the negation of a softgoal, the semantic similarity module has more chance to correctly match the referred goal with the shadow goal than erroneously with the original goal.

At the same time, we add a way to sense clear negation in the goal part of the user-provided expression through defining a list with negation words such as "don't", "doesn't", "not", "nobody", and "couldn't". Such negations are more probable in hardgoals, but can also exist in expressions of softgoals.

The above two sorts of negation information, i.e., whether the matched goal is a shadow goal and whether the natural expression is in a negative form, are passed to the post-processor to create the formal preference, as we demonstrate below.

3.8 Post-Processor

The post-processor accepts as input the matching goal from the goal model, the negation information for the goal as well as the identified preferential strength label and constructs a formal representation as shown in Formula 3.2.

$$PostP = (NegationInfo, Goal)[PreferentialStrengthLabel] \tag{3.2}$$

As an end-to-end example, consider the natural preference expression *"it is quite desirable if the secretary works more"*, provided by the user – who, for the sake of the example, thinks secretaries don't work enough. The Regex will identify a regular expression that matches the particular natural expression, "is (.*?) if (.*)" in our case. The preference part is thus *"quite desirable"* and the goal part is *"the secretary works more"*. The goal part is passed to the UMBC engine along with all goals of the Meeting Scheduling model. The goal that has the highest similarity score is identified as a matching goal, in our case "Increase Labor" with score 0.29, which is a shadow goal of "Reduce Labor". Thus, in the formal preference "Reduce Labor" will occur with a negation, unless a second negation is found in the goal part of the natural expression.

Meanwhile, the preference part is passed to the Preference Identification module which queries the preference repository for key-phrase *"quite desirable"*. The keyword

is not found in the preference repository and it was matched by the semantic similarity with score 50%. If the keyword were *"would be desirable"* instead, an entry would have been found in the repository with 50%. Furthermore, the Negation Identification module does not detect any negation in the goal-part which would cancel the negation already identified by matching a shadow goal. Thus, the post-processor has all the information to construct a formal preference as in Formula 3.3, where 0.5 is the label 50%.

$$PostP = (\neg ReduceLabor)[0.5] \tag{3.3}$$

Considering the goal model of Figure 2.1, in a preference-based reasoning framework, alternatives that contain negative or at least no positive contributions to the goal "Reduce Labor" such as those that involve calling invitees on the phone would return with a higher score. This example is shown from our system in Appendix B.

Chapter Four

Evaluation

4.1 Overview

To evaluate the proposed system, we conducted an exploratory experimental study with human participants. The study has the following objectives:

1. Assess the effectiveness and relevance of the semantic similarity component, i.e., the extent to which participant-supplied natural expressions of intention are matched with the appropriate goal in the goal model.

2. Assess the effectiveness of Regex, i.e., the extent to which natural expressions of preference are correctly matched by one of the provided regular expressions to successfully split them into the goal and preference parts.

3. Assess the scalability and convergence of Regex, i.e., whether subsequent incre-

ments of the number of regular expressions in the Regex repository improve accuracy to a decreasing amount.

4. Explore whether a mapping between preferential expressions to qualitative or quantitative labels is feasible and whether crowdsourced corpora can be the basis for the definition of such mappings.

The study is based on providing our participants goal and preference expressions based on goal models from various domains, asking them to rephrase those expressions, and considering the rephrasing attempts as proxies of potential inputs to our system. We present the experimental design in more detail below.

4.2 Experimental Design

Thirty participants, twenty-four (24) male and six (6) female, are recruited from one undergraduate (16) and one graduate (14) course at York University. Their ages range from 18 to 59 years. Seventeen (17) of them were between 21-29 years. Twelve (12) of the participants are native speakers of English. The demographic details of our study participants can be found in Table 4.1.

The experiment is an on-line instrument, requiring participants to perform a series of five (5) tasks. The tasks in the instrument are based on a particular goal model, chosen from a set of four (4) from the following domains: nursing [14] (23 goals),

Table 4.1: Demographic Details of Participants

Characteristic		**No.**	**%**
Age	18-20	2	7%
	21-29	17	57%
	30-39	9	30%
	40-49	1	3%
	50-59	1	3%
Language	Arabic	1	3%
	English	12	40%
	Mandarin	2	7%
	Persian	3	10%
	Russian	2	7%
	Romanian	2	7%
	Spanish	2	7%
	Other	6	20%
Gender	Male	24	80%
	Female	6	20%
Highest Degree	Bachelor degree	15	50%
	High school degree or equivalent (e.g., GED)	12	40%
	Master degree	2	7%
	Other	1	3%
Field of Degree	Business and Economics	4	13%
	Fine Arts (e.g. Music, Theater, Film)	1	3%
	Health Sciences	1	3%
	Science, Technology and Engineering	22	73%
	Other	2	7%

meeting scheduler [65] (24 goals), car manufacturing [66] (32 goals), and transportation [25] (82 goals). Subjects are distributed to goal models – and therefore domains – in a *between-subjects* design: each participant is randomly assigned to one of the four domains/models.

Before administration of the main tasks, the participants are asked to read a paragraph

describing the domain to which their assigned goal model refers, using phrases taken verbatim from the goal descriptions in the goal model (which, note, the participants never see). The participants also respond to a comprehension question to ensure that the paragraph has been read. The purpose of this is to create a context in which the following tasks are to be understood. Participants perform then the five tasks, as we describe below. Instrument samples can be found in Appendix C.

4.2.1 Task 1

In the first task participants are given five different goals from the goal model and are asked to rephrase them in their own words up to six times. For example, in the meeting scheduling domain, the participants are given the goal "Have Meeting Announced" (taken verbatim from the underlying goal model) and are asked to rewrite it in their own words. To put this exercise in context without exposing participants to unnecessary details, the instrument informs them that their rephrasings will be used to test the natural-language understanding of a hypothetical robotic agent that supports human actors in achieving goals pertaining to the domain. In addition, examples of goals and rephrasings are provided from other unrelated domains (driving, doing laundry, etc.).

4.2.2 Task 2

The second task provides an arbitrary goal called "Achieve X" with a bar indicating a

preference level ranging from 100% to 0%, where 100% is the highest level of importance and 0% is the lowest level of importance. As a graphical aid, an equal percentage of the bar's length is colored. The different levels of importance that are given to the participants to consider are 100%, 75%, 50%, 25%, and 0%. The participants are asked to write a preference for the goal "Achieve X" based on each level of importance, up to six times for each. Two examples are given for 100% and 0%.

4.2.3 Task 3

The third task is a combination of the first and second tasks. The participants are given five different goals, each with a bar indicating a different preference level. The goals are taken from the corresponding goal model. Exactly as in Task 2, for each of the five goals, participants are asked to prepare up to six statements that describe how important each of the goals is, based on a randomly matched level of importance indicated through a numeric label and a colored bar; the levels are, again, 100%, 75%, 50%, 25%, and 0%.

4.2.4 Task 4

The fourth task provides the participants with five different goals taken from the corresponding goal model, each with a different level of importance adopted from Wiegers [67] requirements prioritization scales. The participants are asked to rephrase each goal with its level of importance up to six times. The different levels of importance that are

associated with the goals are *critical, high importance, medium importance, low importance,* and *no importance*.

4.2.5 Task 5

Task five is similar to task four except that (a) all the examples we used only one goal to exclude the effect of the goal on the level of importance and (b) participants were given four levels of importance with one goal adopted this time from Sommerville [68] prioritization scales. The different levels of importance that are associated with the goal are *absolutely essential, important, would be nice,* and *unnecessary*. Again, the participants were asked to rephrase this goal with the different levels of importance up to six times.

4.3 Results

We now turn our focus to the results, based on the evaluation objectives set out above.

4.3.1 Precision of Semantic Similarity Component

To measure the precision of the semantic similarity component of the proposed system (Objective 1) we focus on the results acquired through Task 1. Recall that these results are sets of expressions that constitute participant-provided rephrasing of goals in the goal model. Thus, we collect these expressions, supply them to the semantic similarity component and measure for how many of them the system is able to correctly identify the

original goals, also correctly handling possible negations. As we saw, for each input, the system assigns a similarity measure to each goal in the goal model. The total number of rephrased goals provided by the participants for all the domains was 577. Tables 4.2, 4.3, 4.4, and 4.5 show the results for the Nursing, Meeting Scheduler, Car Manufacturer, and Transportation domains respectively. Each table presents the total number of sentences entered for a specific goal where the original goal had the highest similarity measure (1^{st}), the second highest similarity measure (2^{nd}), the third highest similarity measure (3^{rd}), the fourth highest similarity measure (4^{th}), the fifth highest similarity measure (5^{th}), the number of sentences that were not recognized within the top five similar goals and the percentage for each accordingly. We observe that in all domains the original goal is identified more than half of the times, while it exists in the top 3-5 candidates in the vast majority of times.

Table 4.2: Matching Goal Results for Nursing Domain

No.	**Goal**	**Total**	**1^{st}**	**2^{nd}**	**3^{rd}**	**4^{th}**	**5^{th}**	**Not in top 5**	**% 1^{st}**	**% 2^{nd}**	**% 3^{rd}**	**% 4^{th}**	**% 5^{th}**	**% Not in top 5**
1	Nurse responded to the call	29	19	1	2	1	0	6	65.52%	3.45%	6.90%	3.45%	0.00%	20.69%
2	Nurse walks to the nursing station	23	22	1	0	0	0	0	95.65%	4.35%	0.00%	0.00%	0.00%	0.00%
3	System notifies the nurse through speakers	27	22	2	2	0	0	1	81.48%	7.41%	7.41%	0.00%	0.00%	3.70%
4	Avoid nurse disturbance	29	19	2	0	1	1	6	65.52%	6.90%	0.00%	3.45%	3.45%	20.69%
5	Patient feels cared for	30	13	3	0	1	0	13	43.33%	10.00%	0.00%	3.33%	0.00%	43.33%
	Total	**138**	**95**	**9**	**4**	**3**	**1**	**26**	**68.84%**	**6.52%**	**2.90%**	**2.17%**	**0.72%**	**18.84%**

Table 4.3: Matching Goal Results for Meeting Scheduler Domain

No.	Goal	Total	1st	2nd	3rd	4th	5th	Not in top 5	% 1st	% 2nd	% 3rd	% 4th	% 5th	% Not in top 5
1	Have meeting scheduled	35	29	4	0	1	1	0	82.86%	11.43%	0.00%	2.86%	2.86%	0.00%
2	Book the meeting	30	8	0	0	13	4	5	26.67%	0.00%	0.00%	43.33%	13.33%	16.67%
3	Find suitable slot	29	10	7	0	0	1	11	34.48%	24.14%	0.00%	0.00%	3.45%	37.93%
4	Have meeting announced	29	18	3	4	1	1	2	62.07%	10.34%	13.79%	3.45%	3.45%	6.90%
5	Request constraints by email	28	11	6	7	2	1	1	39.29%	21.43%	25.00%	7.14%	3.57%	3.57%
	Total	**151**	**76**	**20**	**11**	**17**	**8**	**19**	**50.33%**	**13.25%**	**7.28%**	**11.26%**	**5.30%**	**12.58%**

Table 4.4: Matching Goal Results for Car Manufacturer Domain

No.	Goal	Total	1st	2nd	3rd	4th	5th	Not in top 5	% 1st	% 2nd	% 3rd	% 4th	% 5th	% Not in top 5
1	Improve car service	32	30	2	0	0	0	0	93.75%	6.25%	0.00%	0.00%	0.00%	0.00%
2	Reduce raw materials costs	30	29	1	0	0	0	0	96.67%	3.33%	0.00%	0.00%	0.00%	0.00%
3	Reduce operating costs	29	23	3	3	0	0	0	79.31%	10.34%	10.34%	0.00%	0.00%	0.00%
4	Keep labour costs low	30	23	3	0	0	3	1	76.67%	10.00%	0.00%	0.00%	10.00%	3.33%
5	Lower gas price	29	2	8	9	9	0	1	6.90%	27.59%	31.03%	31.03%	0.00%	3.45%
	Total	**150**	**107**	**17**	**12**	**9**	**3**	**2**	**71.33%**	**11.33%**	**8.00%**	**6.00%**	**2.00%**	**1.33%**

Table 4.5: Matching Goal Results for Transportation Domain

No.	Goal	Total	1st	2nd	3rd	4th	5th	Not in top 5	% 1st	% 2nd	% 3rd	% 4th	% 5th	% Not in top 5
1	Manage financial budget	29	25	1	1	0	0	2	86.21%	3.45%	3.45%	0.00%	0.00%	6.90%
2	Transport users to destination	26	14	4	2	3	2	1	53.85%	15.38%	7.69%	11.54%	7.69%	3.85%
3	Improve transport services	31	24	5	1	0	0	1	77.42%	16.13%	3.23%	0.00%	0.00%	3.23%
4	Cover service costs	26	9	3	8	2	0	4	34.62%	11.54%	30.77%	7.69%	0.00%	15.38%
5	Repair means of transport	26	17	1	1	0	4	3	65.38%	3.85%	3.85%	0.00%	15.38%	11.54%
	Total	**138**	**89**	**14**	**13**	**5**	**6**	**11**	**64.49%**	**10.14%**	**9.42%**	**3.62%**	**4.35%**	**7.97%**

Table 4.6 sums the result for all the domains and shows the percentage for each domain where the original goal had the highest similarity measure (column 2), was within the top three similarity measures (column 3), or within the top five similarity measures (column 4).

Table 4.6: Matching Goal Results

Domain	**First Match**	**Match in Top 3**	**Match in Top 5**
Nursing	68.84%	78.26%	81.16%
Car Manufacturer	71.33%	90.67%	98.67%
Meeting Scheduler	50.33%	70.86%	87.42%
Transportation	64.49%	84.06%	92.03%

The cases in which the original goal is not given a high enough similarity measure are often due to issues pertaining to the particular goal model or the experiment. One issue is semantic similarities that exist within the goal model itself. The nursing goal model, for example, contains both goals "Nurse responded to the call" and "Nurse talked to the patient", referring, however, to different things. In the experiment, we asked the participants to rephrase the goal "Nurse responded to the call", and the participants naturally rephrased it by often writing phrases more similar to "Nurse talked to the patient".

In addition, other statements were appropriate to be matched with more than one goal because the rephrased goal, i.e., the natural language statement entered by the participants, combines more than one goal. For example, the goal "Patient feels cared for" was rephrased by a participant "Patient is happy and feels cared for". The highest matching goal for this statement was the goal "Happy patient" with a value of 0.79, while the second matching goal was "Patient feels cared for" with a value of 0.76. These cases were found the most in the nursing and meeting scheduler domains, more examples of

Table 4.7: Semantic Similarity for each Domain

Domain	**Semantic Similairty**
Nursing	0.40
Car Manufacturer	0.22
Meeting Scheduler	0.19
Transportation	0.12

such cases are shown in Appendix D.

Moreover, to quantitatively measure the similarity for the domains that had more phenomena of semantically similar goals, we calculated the average semantic similarity between the goals within each domain, i.e., calculating the semantic similarity of each goal with each other goal in the goal model. Table 4.7 shows the semantic similarity score for each domain, as we can see the nursing domain contains the highest similarity score which seems to support the hypothesis that high levels of internal similarity is detrimental to precision.

Another factor to be noted is the native language of the participants: for the nursing domain we have only two native speakers of English, while for the car manufacturing – the highest results among the domains – contains four native speakers of English, which is the highest number of native speakers among the domains as can be seen in Table 4.8.

Table 4.8: Participants Native Language in Domains

Domain	**No. of Participants**		**Total**
	English	*Non English*	
Meeting Scheduler	3	5	8
Car Manufacturer	4	3	7
Transportation	3	5	8
Nursing	2	5	7
Total	**12**	**18**	**30**

4.3.2 Regular Expressions: Precision and Scalability

With regards to the Regex module, we are firstly interested in the precision of the regular expressions, i.e., how well they identify the preference part from the goal part (Objective 2). In addition, since the effectiveness of this component depends on the choice and number of regular expressions in the repository, we also measure whether it is scalable, i.e., whether there is a minimum number of regular expressions that allow for good precision (Objective 3). To assess these we utilize the results of Tasks 2, 3, 4, and 5.

We use the results of Task 2 to "train" the Regex component and assess whether this training converges to a satisfactory precision. Training here is a manual process of preparing the appropriate regular expressions and adding them to the repository. We work as follows. We begin with twenty six (26) regular expressions which were defined before any training process. Then, we start training Regex by adding rules to the Regex

repository based on user-supplied statements that could not be split with the current state of the repository.

More specifically, through Task 2 participants provide us with a total of 540 training/testing expressions. We divide the expressions into five (5) almost equally sized blocks. Each block is first tested. Then, expressions in the block which fail the test (don't split) despite being legitimate expressions of preference, are used as a basis to manually construct new rules and enrich the regular expression repository. Then we move on to the next block and repeat the same testing-enrichment process.

As it is clearly shown in Figure 4.1 the number of new regular expressions that need to be added in each cycle is decreasing with each new block of training. Thus, the first block introduced twenty four (24) new regular expressions, while the last block introduced four (4) new regular expressions. This offers us evidence that, in practice, the

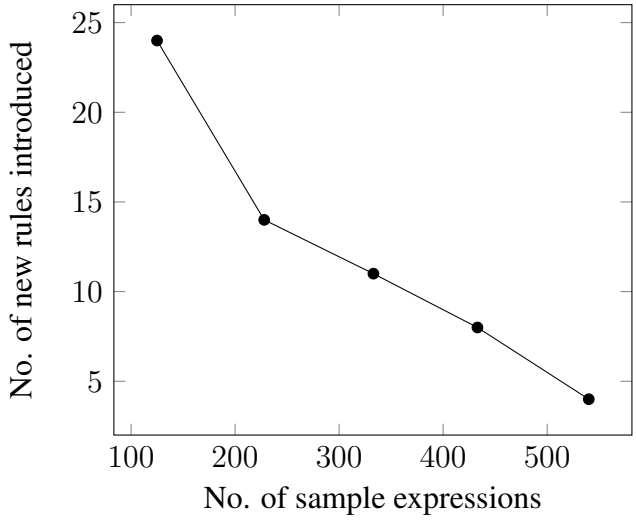


Figure 4.1: New Rules Introduced vs. Examined Expression Samples

Regex may not need perpetual enrichment of its repository, but instead reach an adequate level of precision after a solid initial training investment.

But what is the overall precision in terms of whether all phrases that should be split are split correctly? To assess precision of the splitting process, we performed 10-fold cross validation using the results of Task 2. Specifically, we partition the data into ten equal folds, each fold containing three participants and their natural language statements. We manually train the Regex for 9 of the folds and test the result with the 10th. The precision obtained from this exercise is 92%. This number corresponds to the number of expressions that were successfully split, divided by the total number of expressions that were split (i.e., # true positives / (# true positives + # false positives)). In terms of recall, the number of expressions that were successfully split over the total number of expressions that should have successfully split (i.e., # true positives / (# true positives + # true negatives)), was 84%. Details of each fold in terms of recall and precision can be found in Appendix E and Appendix F respectively.

To further assess precision we utilized the data from Task 3, Task 4, and Task 5. In Task 3, we have 476 statements entered for all domains, which we entered to our system, after training the latter with the results of Task 2. Forty-four (44) of the expressions were incorrectly split into goal and preference components and another fifty (50) statements could not be split with the defined regular expression rules, although they should have. Hence 89.7% (382/(382+44)) of the statements that the system split were a correct split,

Table 4.9: Task 3 Analysis

Domain	No. of Statements	Incorrect Split	Could not Split	Correct Split
Meeting Scheduler	131	9	9	113
Car Manufacturer	116	10	12	94
Transportation	105	10	11	84
Nursing	124	15	18	91
Total	**476**	**44**	**50**	**382**
Percentage	**100%**	**9%**	**11%**	**80%**

while the system was able to correctly split 80.3% (382/476) of all the statements it should have split. Table 4.9 shows the result for Task 3 and the percentage for each column over to the total number of statements.

For Task 4, the number of entered statements for all the domains were 476, eighty (80) of these statements were incorrectly split into goal and preference, while thirty-eight (38) statements could not be split with the defined regular expressions. The precision, i.e., the statements that the system split and were correct, is 81.7% (358/(358+80)), while the recall, i.e., the statements that the system split of all the statements entered, is 75.2% (358/476). Details for the results of Task 4 can be found in Table 4.10.

While for Task 5, 365 statements were entered by the participants for all domains, twenty-eight (28) of them were incorrectly split into goal and preference component and forty-seven (47) of the statements could not be split with the current regular expressions. Hence, the precision is 91.2% (290/(290+28)) and the recall is 79.5% (290/365). Table

Table 4.10: Task 4 Analysis

Domain	No. of Statement	Incorrect Split	Could not Split	Correct Split
Meeting Scheduler	128	27	12	89
Car Manufacturer	119	21	5	93
Transportation	117	4	6	107
Nursing	112	28	15	69
Total	**476**	**80**	**38**	**358**
Percentage	**100%**	**17%**	**8%**	**75%**

4.11 describes the results for Task 5. Note that the statement that could not split also contains some statements that are not applicable, i.e., don't have a clear goal and preference parts.

In Tasks 3, 4, and 5 we also measured the semantic similarity for the identified goal part with all the goals from the goal model to find the most similar goal. In Task 3 the semantic similarity component specifically matched only fourteen (14) of the identified

Table 4.11: Task 5 Analysis

Domain	No. of Statement	Incorrect Split	Could not Split	Correct Split
Meeting Scheduler	108	5	24	79
Car Manufacturer	85	9	5	71
Transportation	83	2	7	74
Nursing	89	12	11	66
Total	**365**	**28**	**47**	**290**
Percentage	**100%**	**8%**	**13%**	**79%**

goal parts into wrong goals in the goal model, whereas the rest (97%) were matched to the correct goals, while in Task 4 the semantic similarity incorrectly matched thirty-one (31) goals into wrong goals, while the rest (93%) were matched correctly and in Task 5 the semantic similarity component mismatched nine (9) goals into wrong goals, while the rest of the goals (98%) were matched correctly. Note that the goals that are matched with the semantic similarity are part of the statements that were correctly split. This higher success rate in these tasks compared to Task 1, is largely due to the fact that participants did not choose to rephrase the goals as they formulated their preferences. The number of goals that were mismatched for each domain can be found in Table 4.12.

Table 4.12: Goal Mismatch

Domain	**Task 3**		**Task 4**		**Task 5**	
	No. of Statements	*Goal Mismatch*	*No. of Statements*	*Goal Mismatch*	*No. of Statements*	*Goal Mismatch*
Meeting Scheduler	131	4	128	13	108	2
Car Manufacturer	116	2	119	9	85	0
Transportation	105	0	117	5	83	0
Nursing	124	8	112	4	89	7
Total	**476**	**14**	**476**	**31**	**365**	**9**

By testing the regular expression component with the previous tasks we found that (a) the regular expression component doesn't need a constant addition and enrichment to it's repository and that it actually converge and (b) the regular expression component results in a high precision and recall for all tasks, the precision for all the tasks was higher than 81% while the recall was higher than 75%.

4.3.3 Preferential Strength Labeling: Feasibility and Effectiveness

Our final evaluation goal asks whether a mapping between natural language expressions of preference and quantitative or qualitative labels is possible (Objective 4). In particular we ask whether a crowd can help us form different corpora of preference expressions, each associated with a distinct preferential strength whether its quantitative (e.g., 100%, 50%, etc.) or qualitative (e.g., critical, high importance, etc.).

To evaluate this, we utilize the results of Task 2 in order to develop the sample corpora and Task 3 to evaluate them. Recall that in these tasks, participants construct preference expressions based on five different preferential strengths that were given to them. For each task these resulted in five different sets of participant-supplied expressions of preference (i.e., five different corpora), each associated with a different strength. We first measure the overlap between the preference parts that were extracted from these sets using the data from Task 2. The overlap is calculated as follows. Each participant-supplied natural expression in the sets is associated with a preference key-phrase that was identified during splitting. The overlap between two corpora is then the number of preference key-phrases that are referenced by both sets divided by the number of preferences referenced to by either set.

Table 4.13 depicts the overlap for each pair of expression sets for Task 2 divided by the total number of preference key-phrases referenced to by the set mentioned in the

row. For example, the 9% cell mentioned under "75%" set in the column is the result of dividing the number of preference key-phrases that were found under "75%" set with the total number of the set mentioned by the row (6/67). As we see, at all times the overlap is below 13% with adjacent sets in terms of preferential strength (e.g., 100% and 75% or 25% and 0%) exhibiting the highest overlap while non-adjacent sets show an overlap that does not exceed 3%. This seems to suggest that natural language expressions of preferential strength for each of those labels are reasonably distinct.

Table 4.13: Intersection between Preferences in our Corpus

	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	67	100%	9%	1%	0%	0%
75%	64	9%	100%	9%	3%	0%
50%	72	1%	8%	100%	13%	1%
25%	71	0%	3%	13%	100%	1%
0%	67	0%	0%	1%	1%	100%

In Figure 4.2 we display the preferences that were used the most by the participants to construct our corpora in Task 2, i.e., the preferences with the highest support, along with the category for each preference. We can see that the most used preferences were *"extremely important"* and *"important"* which were mentioned in our corpora 13 times.

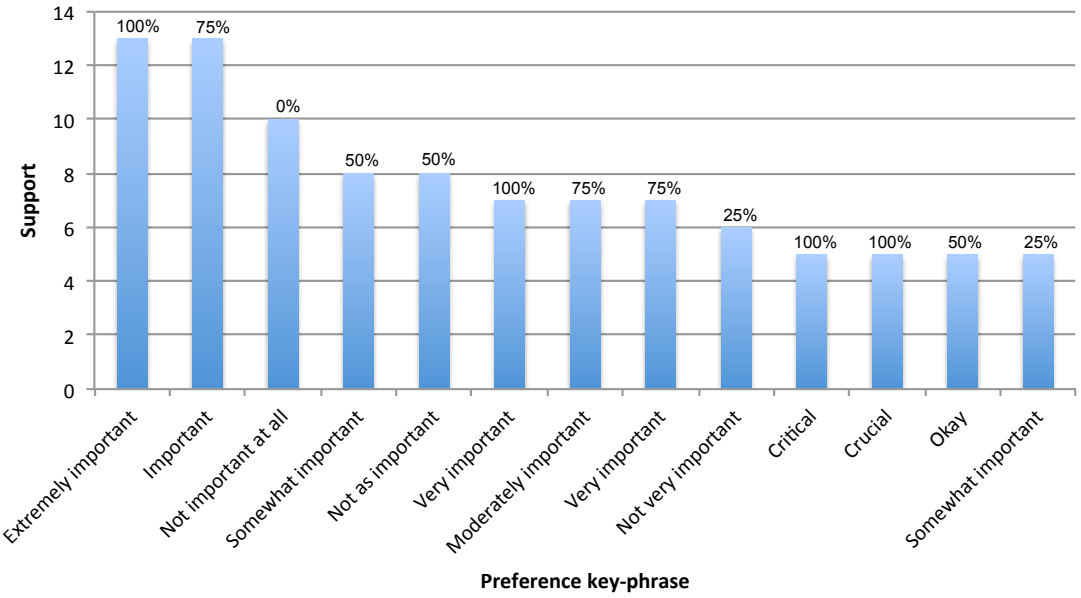


Figure 4.2: Usage of Preference Key-Phrase

4.3.3.1 Quantitative Labeling

Given these results, we went on to use the data from Task 2 as a classifier for the expressions elicited in Task 3. As we saw, in the latter task, each elicited expression from Task 3 is also associated with a 100%, 75%, etc. preferential strength label by user input, and also triggers a specific regular expression in the Regex repository and a preference key-phrase in the preference key-phrase repository.

With respect to preference parts in the preference repository, a total of 315 distinct preference key-phrases are identified from Task 2. When testing with the results of Task 3, 208 of the input natural expressions lead to a pre-existing preference key-phrase respectively, meaning that the corresponding preferential strength labels are taken directly

from the repository. The remaining, however, were not found in the repository, meaning that for those preference parts semantic similarity is used to identify preferential strength.

Consider now the 208 of the total 382 natural expressions of Task 3 that were successfully matched with an existing preference keyword repository (which was developed using data from Task 2). In Table 4.14 each cell represents what proportion of the natural expressions classified by the participants under the label indicated by the row was actually matched by the system to the category indicated by the column. Thus 96% of the responses in Task 3 that were classified by participants under preferential strength label "100%" were also recognized by the system as belonging to the "100%", as inferred by the label of the triggered preference key-phrase. But 8% of the same responses were classified under "75%" (as well). Note here that rows and columns do not necessarily add up to 100% because in case of a draw in support, expressions can be classified to more than one categories. Details for each domain results can be found in Appendix G.

The same analysis was done for the 174 preferences that were new to our corpus and

Table 4.14: Task 3 - Detect Preference Category based on Repository

		The system responses				
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	51	96%	8%	0%	0%	0%
75%	42	29%	81%	7%	0%	0%
50%	32	3%	19%	72%	13%	3%
25%	39	0%	3%	23%	74%	5%
0%	44	2%	0%	0%	23%	75%

Task 3 responses

Table 4.15: Task 3 - Detect Preference Category based on Semantic Similarity

Task 3 responses	Total	The system responses: 100%	75%	50%	25%	0%
100%	32	28%	38%	16%	6%	13%
75%	34	15%	41%	21%	6%	18%
50%	41	12%	17%	32%	15%	24%
25%	32	3%	6%	13%	41%	38%
0%	33	6%	3%	12%	12%	67%

were, hence, matched using the semantic similarity technique as shown in Table 4.15. As above, we compare the strength label to which the system classifies the natural expression, with the label under which the participants provide the expression. Thus, Table 4.15 shows again what proportion of the natural expressions classified by the participants under the label indicated by the row was actually matched by the system to the category indicated by the column, using semantic similarity this time. Note that two of the preferences when matched using the semantic similarity component did not provide a meaningful response, which we attribute to a fault within the STS service that resulted in eliminating them from the analysis. Appendix H shows the results for each domain based on the semantic similarity.

We observe that, in both cases, although variability naturally exists, there is concentration of the highest frequencies in the diagonal (which represents absolutely consistent responses between Tasks 2 and 3) which diminishes as we depart from the diagonal, i.e., as inconsistency level increases.

4.3.3.2 Qualitative Labeling

Task 4 Recall in Task 4 we provide the participants with five different goals each of them associated with qualitative levels of importance such as "*critical*" and "*high importance*" and the users were asked to rephrase these levels of importance with the given goal.

In Task 4 we aim to measure whether a mapping between natural language of preferences to a qualitative labeling is feasible. To do this, we preformed 2-fold cross validation for the preferences collected from the participants. Each fold contains 15 participants and their natural language statements that contains preferences. We train the system (using natural language expressions to construct the preference repository) with one fold and test it (using the remaining natural language expressions) with the other.

The training process here is performed by providing the system with the natural language expressions of 15 participants for each fold. The system matches these statements with the corresponding regular expression (constructed from Task 2) that split them into goal and preference parts. The preferences from the statements that were correctly split into goal and preference parts were used to construct the sets. Thus, the participants provide us with 232 natural language expressions for the first fold and 244 for the second fold. 158 and 200 of these expressions were split correctly into goal and preference parts for the first fold and the second fold, respectively. The preferences for these expressions

were used to construct the corpora.

After training the system with the training folds, we turn into the evaluation process where we used the natural language expressions from the remaining 15 participants in each fold to evaluate the repository that was developed earlier. The remaining participants provide us with 244 statements for the first fold and 232 for the second fold, 200 and 158 of these statements were correctly split into goal and preference component by our regular expression for the first fold and the second fold, respectively. 75 and 62 of these statements contain preferences that existed in our repository while the remaining 125 and 96 preferences were new to our repository for fold one and two, respectively.

As we have done in the quantitative case, we compare the input of the test users, with the output of our system. Thus, for the 137 (75+62) preferences that were found in the system, Table 4.16 shows the percentage of each cell that express the preferences given by the participants under the label indicated by the row is recognized by our system under the label indicated by the column. For example, 19% of the preferences specified by

Table 4.16: Task 4 - Detect Preference Category based on Repository

The system responses

Task 4 responses

	Total	Critical	High importance	Medium importance	Low importance	No importance
Critical	33	48%	52%	0%	0%	0%
High importance	33	21%	76%	3%	0%	0%
Medium importance	26	0%	12%	77%	15%	0%
Low importance	24	0%	0%	17%	79%	13%
No importance	21	0%	0%	0%	19%	95%

the participants under “no importance” was recognized by the system under “low importance”, while 95% of the same preferences were recognized under “no importance”. The same as in Task 3, the columns and rows don’t always add up to 100% because of the possibility to have draw in support. Details for each fold can be found in Appendix I.

For the 221 (125+96) statements whose preference was new to our system, we did the same analysis using the semantic similarity component. The analysis here is done by calculating the average semantic similarity for the preference with each of the preferences under the 5 different sets in our corpora “critical”, “high importance”, etc. and choosing the set with the highest similarity score. Table 4.17 shows the percentage of the preferences specified by the user (rows) is recognized by our system under the preferences label (columns). Note that out of the 221 preferences there were four preferences, i.e., two for each fold, that the semantic similarity component did not provide a meaningful response which resulted in eliminating them from the analysis. The results for each fold can be found in Appendix J.

Table 4.17: Task 4 - Detect Preference Category based on Semantic Similarity

		The system responses				
	Total	**Critical**	**High importance**	**Medium importance**	**Low importance**	**No importance**
Critical	43	9%	44%	16%	5%	26%
High importance	46	15%	54%	2%	24%	4%
Medium importance	44	7%	27%	23%	14%	30%
Low importance	41	2%	7%	10%	41%	39%
No importance	43	7%	5%	0%	51%	37%

Task 4 responses

Although the highest frequencies are not always in the diagonal in regard to the semantic similarity results in this task, we can observe that most of the time the diagonal has the highest frequency or the second highest frequency.

The relation between qualitative and quantitative labels Furthermore to understand the relationship between qualitative and quantitative labels, we also used the data collected from Task 2 as a classifier for the expressions obtained in Task 4 to see whether a mapping between qualitative and quantitive labeling of preferences is feasible. Here we used all the statements provided by the participants in Task 4 and test them with Task 2 repository. As we mentioned earlier, the participants provide us with a total of 476 statements 358 of which were correctly split into goal and preference component. From the 358 statements, 209 statements the preferences were actually found in our system. In Table 4.18 each preference expressed by the participants under the given preference label (rows) is recognized in our system under the preferential strength label (columns). Thus 94% of the preferences specified by the participants under "critical" were recognized by the system under the "100%" preferential strength label, while 19% of the same preferences were recognized under "75%" label. The same here, the columns and rows don't always add up to 100% because of the possibility to have draw in support. The results for each domain can be found in Appendix K.

For the 149 statements that the preference was new to our system, we did the same

Table 4.18: Task 4 - Detect Preference Category based on Task 2 Repository

The system responses

Task 4 responses

	Total	100%	75%	50%	25%	0%
Critical	52	94%	19%	0%	0%	0%
High importance	49	57%	59%	0%	0%	0%
Medium importance	32	0%	22%	63%	9%	6%
Low importance	33	0%	0%	15%	85%	3%
No importance	43	0%	0%	5%	21%	74%

analysis using the semantic similarity component but here we handle all the phrases under each set as one phrase and then calculate the semantic similarity for each set and choose the highest set. Note here that one of the preferences did not provide a meaningful response by the semantic similarity component which resulted in eliminating it from the analysis. Recall in the previous evaluation where we used training data from Task 4 as our repository, we had four unknown preferences while here we have only one. This is because the other unknown preferences were found and matched here by the repository from Task 2.

Table 4.19 shows the percentage of the preferences specified by the user (rows) is recognized by our system under the preferences label (columns). There seems to be some correspondence between qualitative and quantitative labels as the diagonal again contains the highest or second highest overlap. Details for each domain can be found in Appendix L.

Table 4.19: Task 4 - Detect Preference Category based on Task 2 Semantic Similarity

Task 4 responses	Total	The system responses: 100%	75%	50%	25%	0%
Critical	27	33%	33%	7%	15%	11%
High importance	27	37%	44%	11%	4%	4%
Medium importance	37	11%	57%	19%	3%	11%
Low importance	36	0%	14%	22%	25%	39%
No importance	21	14%	5%	10%	24%	48%

Task 5 Similar to Task 4, in this task we provide the participants with qualitative levels of importance such as "*absolutely essential*" and "*important*" but this time using Sommerville's requirements prioritization labels as qualitative labels. Another difference is that we use only one randomly chosen goal for all questions in the task. The participants were asked to rephrase the given levels of importance with the goals.

In this task, we follow the same evaluation as in Task 4: we performed 2-fold cross validation for the preferences provided by the participants. Each fold contains statements of 15 participants, one fold was used for training and the other fold for testing.

For the training part, the participants provide us with 171 and 194 natural language expressions that contains preferences for the first fold and the second fold, respectively. 138 of these expressions split correctly into goal and preference parts for the first fold while 152 split for the second fold. The preferences from these expressions were used to construct the corpora.

For the testing part, the remaining participants provide us with 194 and 171 natural language expressions for the first fold and the second fold, respectively. 152 of these expressions split correctly into goal and preference parts for the first fold while 138 of them split correctly for the second fold.

Consider now the 152 and 138 statements that split correctly by the regular expression for fold one and two, respectively. 66 (resp. 53) of these statements contain preferences that were actually matched with an existing preference key-phrase in the repository, i.e., that was constructed from the training fold, while 86 (resp. 85) of these statements contain preferences that were new to our system in regard to the first and second folds, respectively.

As in the previous tasks, we did the same analysis for the 119 (66+53) preferences that existed in our system, Table 4.20 provide us with the percentage for the preferences that were expressed by the participants by the label indicated by the row was found in our

Table 4.20: Task 5 - Detect Preference Category based on Repository

		The system responses			
	Total	**Absolutely essential**	**Important**	**Would be nice**	**Unnecessary**
Task 5 responses: **Absolutely essential**	36	94%	8%	0%	0%
Important	27	15%	85%	0%	0%
Would be nice	24	0%	0%	92%	8%
Unnecessary	32	0%	0%	9%	91%

system under the label indicated by the column. Thus, 94% of the preferences that were specified by the participants under "absolutely essential" were found in our system under "absolutely essential" while 8% of the same preferences were found under "Important". We observe that Sommerville's 4-level labeling offers seemingly better defined corpora than Wiegers's 5-level one. The details for each fold can be found in Appendix M.

Now for the 171 (86+85) preferences that were new to our system, we match them using the semantic similarity component as in the previous task where we calculated the average semantic similarity for the preference with all the preferences in each set and choose the set with the highest similarity score. Note that the semantic similarity component did not provide a meaningful response for four of the preferences, i.e., two for each fold, which resulted in eliminating them from the analysis. Table 4.21 shows the preference category specified by the participants intersect with the category specified by the system. The same observation found in the semantic similarity matching where

Table 4.21: Task 5 - Detect Preference Category based on Semantic Similarity

		The system responses			
	Total	**Absolutely essential**	**Important**	**Would be nice**	**Unnecessary**
Absolutely essential	46	57%	24%	9%	11%
Important	43	53%	28%	5%	14%
Would be nice	40	13%	13%	55%	20%
Unnecessary	38	11%	13%	3%	74%

Task 5 responses

the Sommerville's 4-level labeling offers better results than Wiegers's 5-level labeling. More details for each fold can be found in Appendix N.

The relation between qualitative and quantitative labels Again, in this task we matched the preferences with Task 2 quantitive labels, here we used all the natural language statements (365) provided by the participants in Task 5. From the 365 statements, 290 were correctly split into goal and preference parts, 157 of these preferences were found in our repository while 133 were new to the system. In Table 4.22 we matched the 157 statements that their preferences were found in our repository, i.e., constructed from Task 2, with the matching set from our repository. Thus, 98% of the preferences that were mentioned by the participants under "absolutely essential" were found in our system under "100%" preferential strength label while 4% of these preferences were matched in our system under "75%". The results for each domain can be found in Appendix O.

Table 4.22: Task 5 - Detect Preference Category based on Task 2 Repository

Task 5 responses \ The system responses	Total	100%	75%	50%	25%	0%
Absolutely essential	49	98%	4%	0%	0%	0%
Important	42	38%	62%	2%	0%	0%
Would be nice	17	0%	12%	71%	41%	0%
Unnecessary	49	0%	2%	0%	18%	80%

For the rest of the preferences (133) that were not found in our system, we did the same analysis using the semantic similarity component where we match the preference with all the preferences in each set by treating them as one phrase and then choose the set with the highest similarity score. Note here that one preference did not provide a meaningful response by the semantic similarity component which resulted in eliminating it from the analysis. Recall in the previous evaluation where we used training data from Task 5 as our repository, we had four unknown preferences while here we have only one. This is because the other unknown preferences were found and matched here by the repository from Task 2.

In Table 4.23 we can find the percentage of the preferences specified by the user under the row which was matched by our system with the label indicated by the column. More specified details for each domain can be found in Appendix P.

Table 4.23: Task 5 - Detect Preference Category based on Task 2 Semantic Similarity

		The system responses				
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Task 5 responses: **Absolutely essential**	34	56%	24%	9%	9%	3%
Important	29	24%	45%	14%	3%	14%
Would be nice	48	6%	21%	50%	15%	8%
Unnecessary	21	10%	5%	5%	5%	76%

4.4 Limitations and Threats to Validity

We find the results of the empirical study to be encouraging: semantic similarity via application of the UMBC LSA-WordNet framework is reasonably accurate, regular expressions seem to capture the vast majority of user supplied natural expressions of preference, without, apparently, the need for continuous enrichment, and crowd-based supply of examples has an evident potential to be used for preferential strength classifiers. Nevertheless, as any empirical work, our exploratory experiment is exposed to validity threats. We discuss external and construct validity, which we find particularly relevant in our study and necessitate further work.

External Validity refers to the extent to which our findings are generalizable. The threat becomes present in three ways. First, the participant sample, identified through opportunity sampling, is restricted to students of Information Technology. Furthermore, many of the students (18) do not have English as their first language. As such, generalization hypotheses should probably be restricted to groups with similar features. Second, the domains of the goal models and the familiarity of the participants to them may have an effect to the ease by which consistent expressions of intention and preference are generated. Our four models, for example, offered us some noticeable variability in the results. Although, as we saw, these differences had mostly to do with the construction of the goal model itself rather than the nature of the domain (e.g., the goal model contained

semantically similar yet distinct goals), further experimentation, perhaps with domains of more esoteric vocabularies (e.g., financial) would shed more light on the influence of the domain of choice. Third, the goal models considered are of small-to-medium size. In future experimentation, we may find that larger goal models could impact the precision of the semantic similarity components: larger models, for example, could be more likely to contain semantically similar goals. Noting, of course, that goal models of the size we considered are still useful, one would probably prefer to be reluctant to make any generalization statements for models with far larger sizes.

Construct Validity refers to the appropriateness of the instrument by which we acquire expressions of intention and preference. While in reality such expressions are made by stakeholders when confronted with a problem that concerns them and during performance of a goal-oriented activity (e.g., make a decision, configure, explore), our experiment is restricted to re-phrasing exercises. There is, thus, a possibility that spontaneous expression of intent and preference has different characteristics from what we acquired. Alternative, perhaps more naturalistic designs can be considered in the future to answer this. In addition, targeted evaluation of expressions of negation may also be needed for a more thorough understanding of the effectiveness of the negation identification component.

Nevertheless, one must also note that the presented evaluation may use stricter assumptions than those posed by the context for which the system is envisioned, i.e., a

decision making or configuration tool. For example, users may in practice receive some exposure on suggested ways to phrase a preference, instead of inventing them as they did here, and may eventually be vaguely aware of how the goals are phrased in the goal model and use those phrasings, instead of having to make up their own. In addition, within a decision making tool, natural preferences are not likely to be specified in a one-shot manner, but rather interactively whereby the system initiates a confirmation/clarification dialogue (*"did you mean [...]?"*), in which likely possibilities (e.g., lower ranked goals in the semantic similarity result) are presented for selection. This means that, in practice, the application context might be, to a certain extent, forgiving to imprecision.

Chapter Five

Conclusion

5.1 Summary and Contributions

This thesis presented a system for translating natural language expressions of preference into formal preference specifications to be used for formal reasoning with goal models. The system is based on a combination of regular expressions, statistical semantic similarity, and the development of a corpus-based classifier for preferential strength. Experimental evaluation indicates that both regular expressions and semantic similarity are encouragingly effective, and that developing and using corpora for identification of preferential strength is feasible.

Our contribution lies in three possible areas:

1. We propose an approach that can potentially increase the attainability of preference-based and goal reasoning toolkits using natural language techniques.

2. We propose a way to elicit preferential strengths using the utility of natural language, which may also be usable in more general requirements prioritization contexts.

3. Finally, we extend the previous proposals for preference-based software customization to enable non-technical users to use natural language in system customizations.

5.2 Future Work

For the future we wish to attempt different evaluation approaches, considering more natural and contextualized ways to acquire input from participants (e.g., a real or artificial decision making problem or a specific customization problem). In addition we wish to try alternative technologies for identifying the goal and preference, including parsing and analyzing the sentence [69] to show it's syntactic categorization and define the subject and object which could help in identifying the goal and preference parts.

Moreover, we would like to try more techniques for matching the sentence with the right goal from the goal model by using probabilistic techniques such as Bayesian approach [70, 46]. In applying the Bayesian approach to our problem, the semantic concept for each goal need to be extracted, and then the sentence entered by the user will be probabilistically mapped to the right goal according to the predefined semantic concepts.

Other techniques include even common information retrieval methods e.g., tf-idf where the focus is on the importance of the words appearing in a document based on the frequency that they appear on other documents which in our case will be the goals from the goal model. Finally, techniques for enabling automated learning and improvement of the natural language understanding mechanism, based on its continuous use and feedback by users, are certainly worth consideration for future development.

In practice we aim that our proposed system can be used to build better tools for exploring alternatives in a goal model. An example of such a tool is a goal editor application that allows NL queries where the user/ system analyst can enter the goal model and then start specifying the preferences in natural language. Another variation the application could display the goal model and a prompt asking the user "*What are you interested in?*" and the user will type his preferences while in the background the system will translate these preferences so certain alternatives within the goal model will disappear or become less salient because they are irrelevant and do not match the preferences specified. In addition, the use of our system in a goal oriented configuration system could be useful where an ordinary user can configure the system using natural language expression of preferences. For instance the users can change the configuration of their browser or email client by using natural language that they use for daily communication.

Bibliography

[1] F. Alabdulkareem, N. Cercone, and S. Liaskos, "Goal and preference identification through natural language," in *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, Aug 2015, pp. 56–65.

[2] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Science of Computer Programming*, vol. 20, no. 1-2, pp. 3–50, 1993.

[3] E. S. K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *Proc. of the 3rd IEEE Int. Symposium on Requirements Engineering (RE'97)*, Washington D.C., January 1997.

[4] E. S. K. Yu and J. Mylopoulos, "Understanding "why" in software process modelling, analysis, and design," in *Proceedings of the 16th International Conference on Software Engineering (ICSE'94)*, Sorrento, Italy, 1994, pp. 159–168.

[5] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach," *IEEE Transactions on Software Engi-*

neering, vol. 18, no. 6, pp. 483–497, 1992.

[6] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional Requirements in Software Engineering*. Springer Science & Business Media, 2012, vol. 5.

[7] E. Yu, "Modelling strategic relationships for process reengineering," *Social Modeling for Requirements Engineering*, vol. 11, p. 2011, 2011.

[8] J. Castro, M. Kolp, and J. Mylopoulos, "Towards requirements-driven information systems engineering: the Tropos project," *Information Systems*, vol. 27, no. 6, pp. 365–389, 2002.

[9] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.

[10] D. Amyot and G. Mussbacher, "URN: towards a new standard for the visual description of requirements," in *Telecommunications and beyond: The Broader Applicability of SDL and MSC, Third International Workshop, SAM 2002, Aberystwyth, UK, June 24-26, 2002. Revised Papers*, 2002, pp. 21–37.

[11] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu, "Exploring alternatives during requirements analysis," *IEEE Software*, vol. 18, no. 1, pp. 92 – 96, 2001.

[12] F. Aydemir, P. Giorgini, J. Mylopoulos, and F. Dalpiaz, "Exploring alternative designs for sociotechnical systems," in *Proc. of the 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, May 2014, pp. 1–12.

[13] S. Liaskos, S. McIlraith, S. Sohrabi, and J. Mylopoulos, "Representing and reasoning about preferences in requirements engineering," *Requirements Engineering Journal (REJ)*, vol. 16, pp. 227–249, 2011.

[14] S. Liaskos, S. A. McIlraith, and J. Mylopoulos, "Towards augmenting requirements models with preferences," in *Proc. of the 24th International Conference on Automated Software Engineering (ASE'09)*, Auckland, New Zealand, 2009, pp. 565–569.

[15] S. Liaskos, S. M. Khan, M. Litoiu, M. D. Jungblut, V. Rogozhkin, and J. Mylopoulos, "Behavioral adaptation of information systems through goal models," *Information Systems (IS)*, vol. 37, no. 8, pp. 767–783, 2012.

[16] S. Liaskos, A. Lapouchnian, Y. Wang, Y. Yu, and S. Easterbrook, "Configuring common personal software: a requirements-driven approach." in *Proc. of the 13th IEEE International Requirements Engineering Conference (RE'05)*, Paris, France, 2005.

[17] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese, "UMBC EBIQUITY-CORE: Semantic textual similarity systems," in *Proc. of the Second Joint Conference on Lexical and Computational Semantics*, vol. 1, 2013, pp. 44–52.

[18] J. Horkoff and E. Yu, "A qualitative, interactive evaluation procedure for goal-and agent-oriented models," in *CAiSE Forum*, 2009.

[19] H. Simon, "The sciences of the artificial 2nd edition," 1981.

[20] L. Chung and J. C. S. do Prado Leite, "On non-functional requirements in software engineering," in *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, 2009, pp. 363–379.

[21] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, "The NFR framework in action," in *Non-Functional Requirements in Software Engineering*, ser. International Series in Software Engineering. Springer US, 2000, vol. 5, pp. 15–45. [Online]. Available: http://dx.doi.org/10.1007/978-1-4615-5269-7_2

[22] E. Kavakli and P. Loucopoulos, "Goal driven requirements engineering: Evaluation of current methods," in *Proceedings of the 8th CAiSE/IFIP8*, vol. 1, 2003, pp. 16–17.

[23] H. Luo and D. Amyot, "Towards a declarative, constraint-oriented semantics with a generic evaluation algorithm for GRL," in *Proceedings of the 5th International* i* *Workshop 2011, Trento, Italy, August 28-29, 2011*, 2011, pp. 26–31.

[24] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. S. K. Yu, "Evaluating goal models within the goal-oriented requirement language," *Int. J. Intell. Syst.*, vol. 25, no. 8, pp. 841–877, 2010.

[25] R. Sebastiani, P. Giorgini, and J. Mylopoulos, "Simple and minimum-cost satisfiability for goal models," in *Proc. of the 16th Conference On Advanced Information Systems Engineering (CAiSE'04)*, Riga, Latvia, 2004, pp. 20–35.

[26] R. Grishman, "Information extraction: Techniques and challenges," in *Proc. of the International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology (SCIE '97)*, London, UK, UK, 1997, pp. 10–27.

[27] F. Alkhateeb, J.-F. Baget, and J. Euzenat, "Extending {SPARQL} with regular expression patterns (for querying rdf)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 2, pp. 57 – 73, 2009.

[28] Y. Li, D. Mclean, Z. Bandar, J. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, Aug 2006.

[29] C. T. Meadow, B. R. Boyce, and D. H. Kraft, *Text Information Retrieval Systems*. Academic Press Orlando, 1992, vol. 2.

[30] J. L. McClelland and A. H. Kawamoto, "Mechanisms of sentence processing: Assigning roles to constituents," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models*, Cambridge, MA, 1986, pp. 272–325.

[31] V. Hatzivassiloglou, J. L. Klavans, and E. Eskin, "Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning," in *Proc. of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 203–212.

[32] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[33] C. Burgess, K. Livesay, and K. Lund, "Explorations in context space: words, sentences, discourse," *Discourse Processes*, vol. 25, no. 2-3, pp. 211–257, 1998.

[34] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[35] Stanford, "Stanford webbase project," 2001. [Online]. Available: http://bit.ly/WebBase

[36] L. Han and T. Finin, "Umbc webbase corpus," 2013. [Online]. Available: http://ebiq.org/r/351

[37] Z. S. Harris, *Mathematical Structures of Language*. Wiley, 1968.

[38] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "The detection and classification of non-functional requirements with application to early aspects," in *Proc. of the 14th IEEE International Requirements Engineering Conference*, Sept 2006, pp. 39–48.

[39] ——, "Automated classification of non-functional requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 103–120, 2007.

[40] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology*, vol. 52, no. 4, pp. 436 – 445, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584909001918

[41] T. Breaux and A. Anton, "Analyzing goal semantics for rights, permissions, and obligations," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, Aug 2005, pp. 177–186.

[42] J. H. Weber-Jahnke and A. Onabajo, "Finding defects in natural language confidentiality requirements," in *Proc. of the 17th IEEE International Requirements Engineering Conference, Atlanta, GA*, 2009, pp. 213–222.

[43] H. Yang, A. N. D. Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Speculative requirements: Automatic detection of uncertainty in natural language requirements," in *Proc. of the 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL*, 2012, pp. 11–20.

[44] J. Horkoff and E. Yu, "Finding solutions in goal models: an interactive backward reasoning approach," in *Proc. of the 29th International Conference on Conceptual modeling (ER'10)*, Vancouver, Canada, 2010, pp. 59–75.

[45] ——, "Visualizations to support interactive goal model analysis," in *Proc. of the 5th International Workshop on Requirements Engineering Visualization (REV'10)*, Sept 2010, pp. 1–10.

[46] Y. He, "Goal detection from natural language queries," in *Proc. of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB 2010)*, Cardiff, UK, 2010, pp. 157–168.

[47] H. M. Meng, C. Wai, and R. Pieraccini, "The use of belief networks for mixed-initiative dialog modeling," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 757–773, 2003.

[48] E. Casagrande, S. Woldeamlak, W. L. Woon, H. H. Zeineldin, and D. Svetinovic, "NLP-KAOS for systems goal elicitation: Smart metering system case study," *IEEE Transactions on Software Engineering*, vol. 40, no. 10, pp. 941–956, 2014.

[49] M. Kröll, C. Körner, and M. Strohmaier, "iTAG: Automatically annotating textual resources with human intentions," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 4, 2010.

[50] I. Nunes, S. D. Barbosa, D. Cowan, S. Miles, M. Luck, and C. J. de Lucena, "Natural language-based representation of user preferences," *Interacting with Computers*, vol. 27, no. 2, pp. 133–158.

[51] I. Nunes, S. Miles, M. Luck, S. D. J. Barbosa, and C. J. P. de Lucena, "Decision making with natural language based preferences and psychology-inspired heuristics," *Engineering Applications of AI*, vol. 42, pp. 16–35, 2015.

[52] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo, "Recovering traceability links between code and documentation," *Software Engineering, IEEE Transactions on*, vol. 28, no. 10, pp. 970–983, Oct 2002.

[53] M. Gibiec, A. Czauderna, and J. Cleland-Huang, "Towards mining replacement queries for hard-to-retrieve traces," in *ASE 2010, 25th IEEE/ACM International Conference on Automated Software Engineering, Antwerp, Belgium, September 20-24, 2010*, 2010, pp. 245–254.

[54] H. Köpcke, A. Thor, S. Thomas, and E. Rahm, "Tailoring entity resolution for matching product offers," in *15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings*, 2012, pp. 545–550.

[55] M. de Bakker, F. Frasincar, and D. Vandic, "A hybrid model words-driven approach for web product duplicate detection," in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, C. Salinesi, M. Norrie, and . Pastor, Eds. Springer Berlin Heidelberg, 2013, vol. 7908, pp. 149–161.

[56] D. Vandic, J.-W. Van Dam, and F. Frasincar, "Faceted product search powered by the semantic web," *Decision Support Systems*, vol. 53, no. 3, pp. 425–437, 2012.

[57] K. Lagus and J. Kuusisto, "Topic identification in natural language dialogues using neural networks," in *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue-Volume 2*. Association for Computational Linguistics, 2002, pp. 95–102.

[58] T. Kohonen, "Analysis of a simple self-organizing process," *Biological Cybernetics*, vol. 44, no. 2, pp. 135–140, 1982.

[59] ——, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1–3, pp. 1–6, 1998.

[60] C. Clifton, R. Cooley, and J. Rennie, "Topcat: Data mining for topic identification in a text corpus," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 8, pp. 949–964, Aug 2004.

[61] M. Damashek, "Method of retrieving documents that concern the same topic," May 23 1995, uS Patent 5,418,951.

[62] J. R. Bellegarda and K. E. A. Silverman, "Natural language spoken interface control using data-driven semantic inference," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 267–277, 2003.

[63] D. Jin, M. B. Cohen, X. Qu, and B. Robinson, "PrefFinder: Getting the right preference in configurable software systems," in *Proc. of the ACM/IEEE International Conference on Automated Software Engineering, ASE '14*, Vasteras, Sweden, 2014, pp. 151–162.

[64] S. Liaskos, S. A. McIlraith, S. Sohrabi, and J. Mylopoulos, "Representing and reasoning about preferences in requirements engineering," *Requir. Eng.*, vol. 16, no. 3, pp. 227–249, 2011.

[65] S. Liaskos, S. M. Khan, M. Soutchanski, and J. Mylopoulos, "Modeling and reasoning with decision-theoretic goals," in *Proc. of the 32th International Conference on Conceptual Modeling (ER 2013), Hong-Kong, China*, 2013, pp. 19–32.

[66] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models," in *Proc. of the 21st International Conference on Conceptual Modeling (ER'02)*, London, UK, 2002, pp. 167–181.

[67] K. Wiegers, "First things first: prioritizing requirements," *Software Development*, vol. 7, no. 9, pp. 48–53, 1999.

[68] I. Sommerville and P. Sawyer, *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.

[69] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *Proc. of the 2013 Annual Meeting of the Association for Computational Linguistics*, 2013.

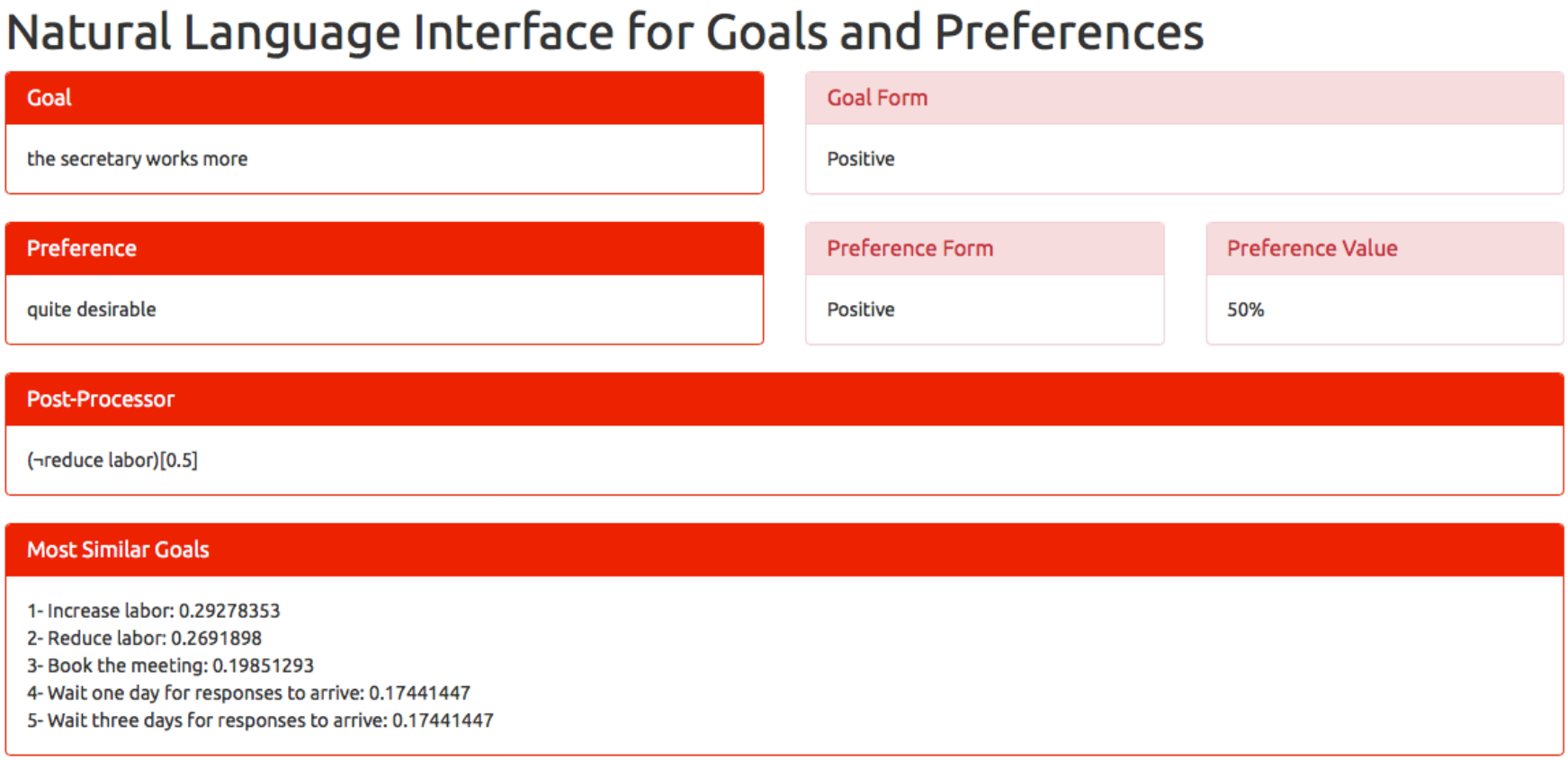
[70] D. Heckerman and E. Horvitz, "Inferring informational goals from free-text queries: A bayesian approach," in *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, 1998, pp. 230–237.

Appendices

A Preference Key-Phrases with Labels and Supports

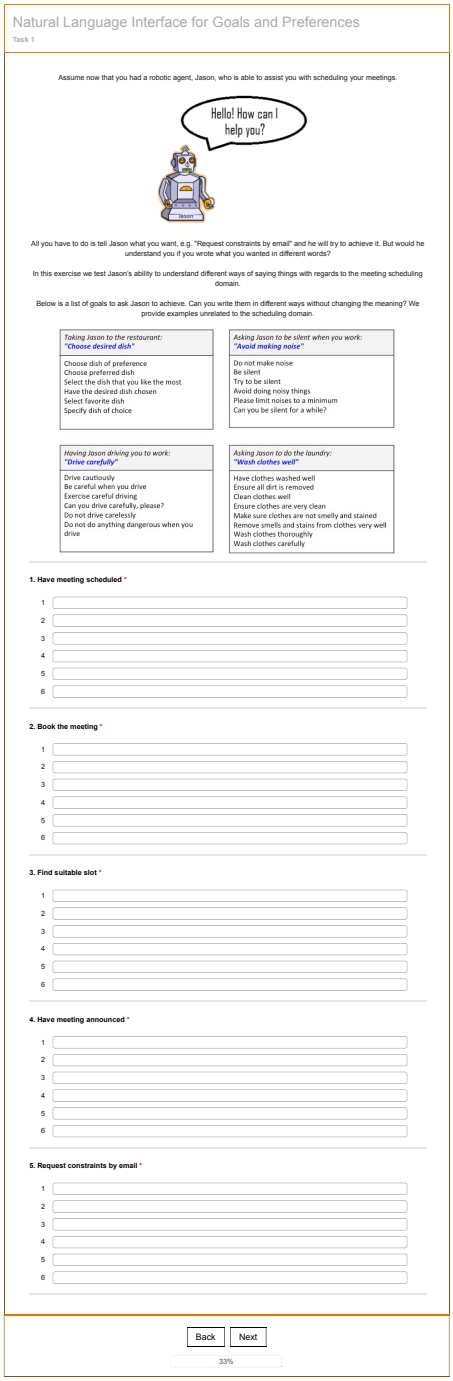
Label	Preference key-phrase	Support
100%	highly important	2
	extremely important	13
	absolutely required	2
	essential	4
	critical	5
75%	important	13
	recommended	2
	moderately important	7
	pretty important	2
	your second priority	1
50%	somewhat important	8
	good	2
	okay	5
	neither important nor unimportant	2
	not as important	8
25%	not very important	6
	low priority	2
	relatively unimportant	2
	not significant	1
	little importance	2
0%	not important at all	10
	not required	3
	lowest priority	3
	not needed	2
	inappreciable	3

B System Output



C Instrument Sample

C.1 Task 1



Natural Language Interface for Goals and Preferences

Task 1

Assume now that you had a robotic agent, Jason, who is able to assist you with scheduling your meetings.

All you have to do is tell Jason what you want, e.g. "Request constraints by email" and he will try to achieve it. But would he understand you if you wrote what you wanted in different words?

In this exercise we test Jason's ability to understand different ways of saying things with regards to the meeting scheduling domain.

Below is a list of goals to ask Jason to achieve. Can you write them in different ways without changing the meaning? We provide examples unrelated to the scheduling domain.

Taking Jason to the restaurant: ***"Choose desired dish"***
Choose dish of preference
Choose preferred dish
Select the dish that you like the most
Have the desired dish chosen
Select favorite dish
Specify dish of choice

Asking Jason to be silent when you work: ***"Avoid making noise"***
Do not make noise
Be silent
Try to be silent
Avoid doing noisy things
Please limit noises to a minimum
Can you be silent for a while?

Having Jason driving you to work: ***"Drive carefully"***
Drive cautiously
Be careful when you drive
Exercise careful driving
Can you drive carefully, please?
Do not drive carelessly
Do not do anything dangerous when you drive

Asking Jason to do the laundry: ***"Wash clothes well"***
Have clothes washed well
Ensure all dirt is removed
Clean clothes well
Ensure clothes are very clean
Make sure clothes are not smelly and stained
Remove smells and stains from clothes very well
Wash clothes thoroughly
Wash clothes carefully

1. Have meeting scheduled *

1
2
3
4
5
6

2. Book the meeting *

1
2
3
4
5
6

3. Find suitable slot *

1
2
3
4
5
6

4. Have meeting announced *

1
2
3
4
5
6

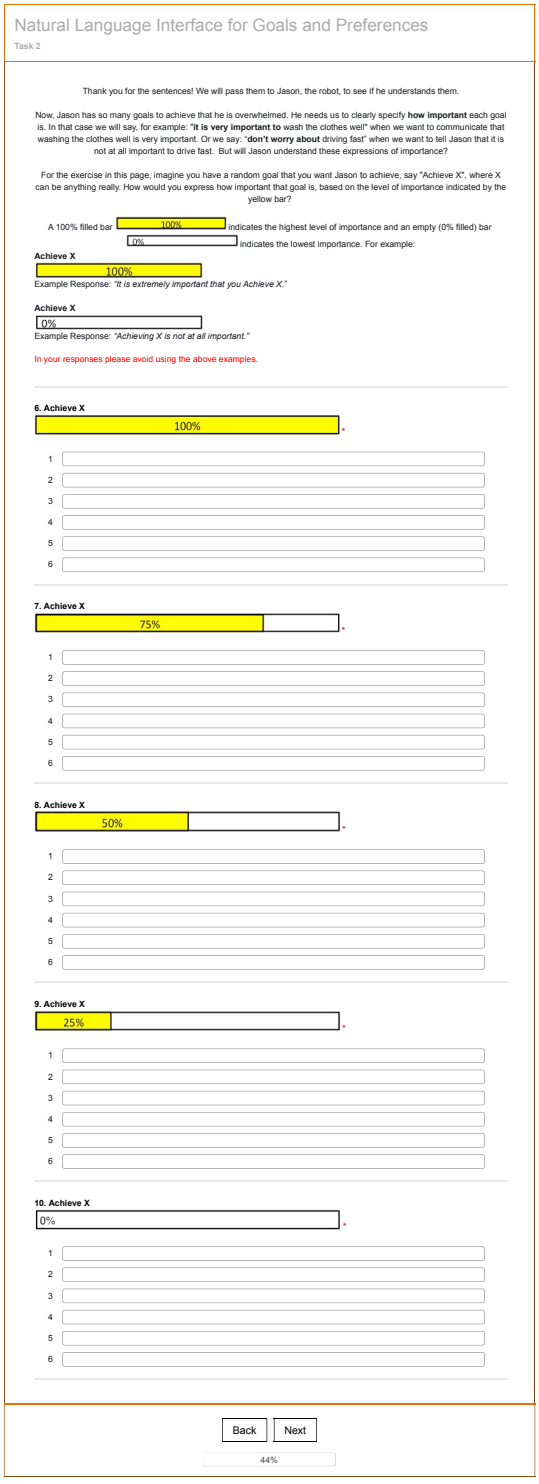
5. Request constraints by email *

1
2
3
4
5
6

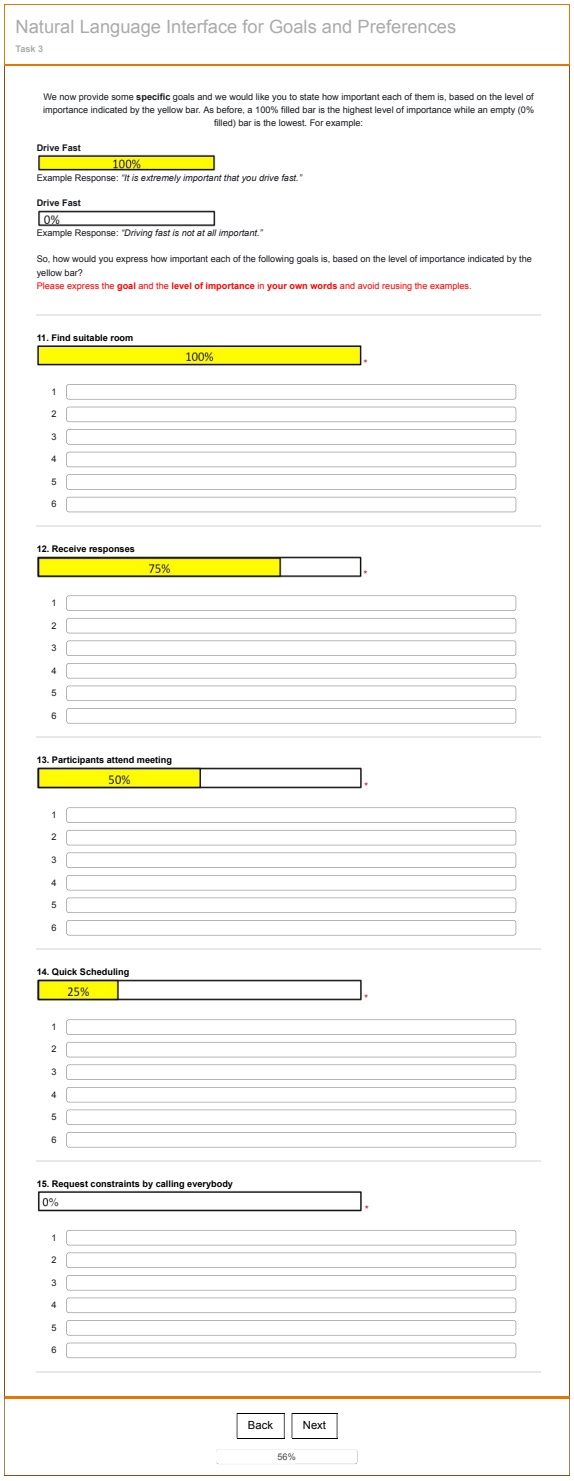
Back Next

33%

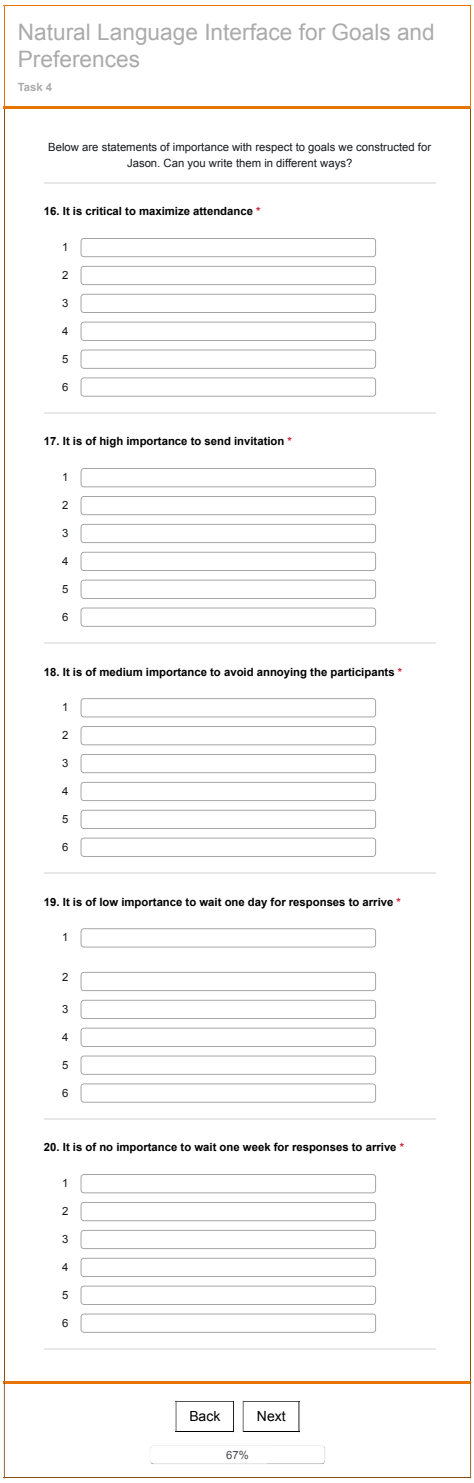
C.2 Task 2


Natural Language Interface for Goals and Preferences
Task 2
Thank you for the sentences! We will pass them to Jason, the robot, to see if he understands them.
Now, Jason has so many goals to achieve that he is overwhelmed. He needs us to clearly specify how important each goal is. In that case we will say, for example: "It is very important to wash the clothes well" when we want to communicate that washing the clothes well is very important. Or we say: "don't worry about driving fast" when we want to tell Jason that it is not at all important to drive fast. But will Jason understand these expressions of importance?
For the exercise in this page, imagine you have a random goal that you want Jason to achieve, say "Achieve X", where X can be anything really. How would you express how important that goal is, based on the level of importance indicated by the yellow bar?
A 100% filled bar 100% indicates the highest level of importance and an empty (0% filled) bar 0% indicates the lowest importance. For example:
Achieve X
100%
Example Response: "It is extremely important that you Achieve X."
Achieve X
0%
Example Response: "Achieving X is not at all important."
In your responses please avoid using the above examples.
6. Achieve X
100%
1
2
3
4
5
6
7. Achieve X
75%
1
2
3
4
5
6
8. Achieve X
50%
1
2
3
4
5
6
9. Achieve X
25%
1
2
3
4
5
6
10. Achieve X
0%
1
2
3
4
5
6
Back
Next
44%

C.3 Task 3


Natural Language Interface for Goals and Preferences
Task 3
We now provide some specific goals and we would like you to state how important each of them is, based on the level of importance indicated by the yellow bar. As before, a 100% filled bar is the highest level of importance while an empty (0% filled) bar is the lowest. For example:
Drive Fast
100%
Example Response: "It is extremely important that you drive fast."
Drive Fast
0%
Example Response: "Driving fast is not at all important."
So, how would you express how important each of the following goals is, based on the level of importance indicated by the yellow bar?
Please express the goal and the level of importance in your own words and avoid reusing the examples.
11. Find suitable room
100%
12. Receive responses
75%
13. Participants attend meeting
50%
14. Quick Scheduling
25%
15. Request constraints by calling everybody
0%
1
2
3
4
5
6
Back
Next
56%

C.4 Task 4



Natural Language Interface for Goals and Preferences

Task 4

Below are statements of importance with respect to goals we constructed for Jason. Can you write them in different ways?

16. It is critical to maximize attendance *

1
2
3
4
5
6

17. It is of high importance to send invitation *

1
2
3
4
5
6

18. It is of medium importance to avoid annoying the participants *

1
2
3
4
5
6

19. It is of low importance to wait one day for responses to arrive *

1
2
3
4
5
6

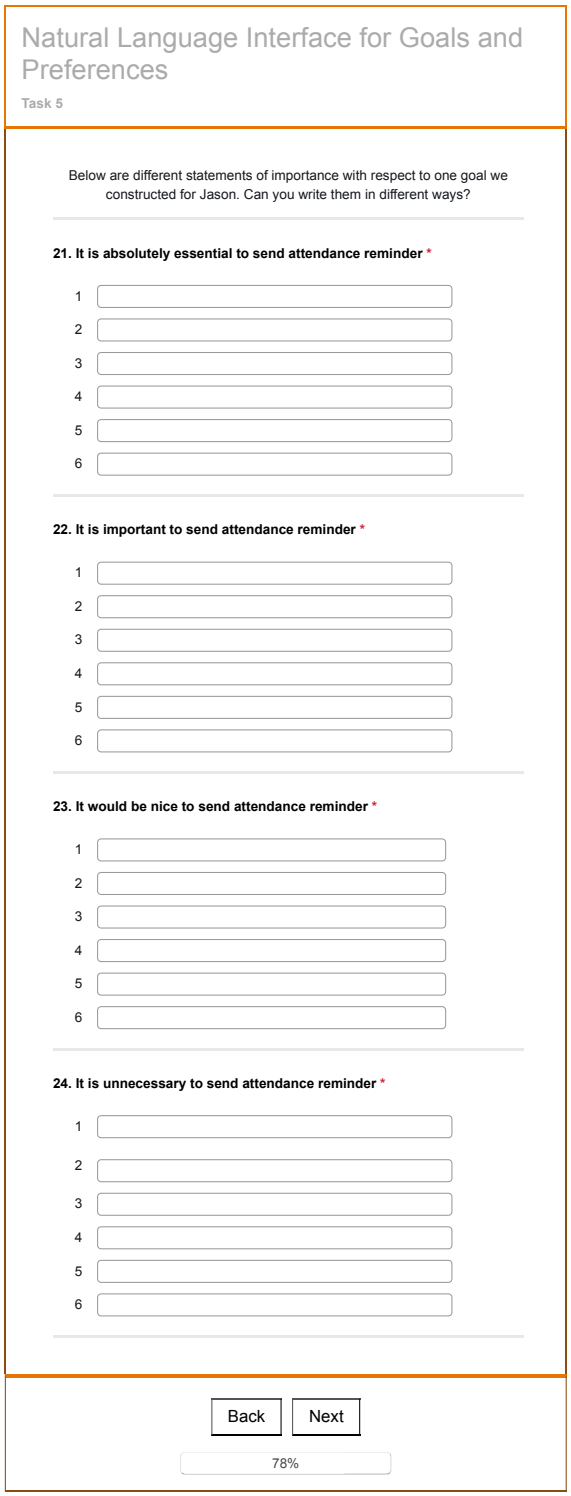
20. It is of no importance to wait one week for responses to arrive *

1
2
3
4
5
6

Back Next

67%

C.5 Task 5

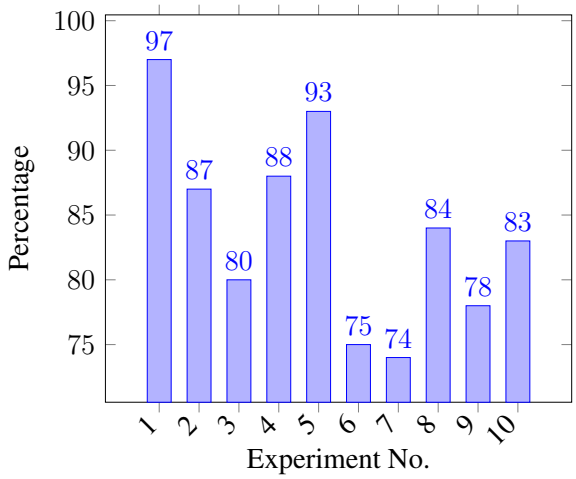

Natural Language Interface for Goals and Preferences
Task 5
Below are different statements of importance with respect to one goal we constructed for Jason. Can you write them in different ways?
21. It is absolutely essential to send attendance reminder *
1
2
3
4
5
6
22. It is important to send attendance reminder *
1
2
3
4
5
6
23. It would be nice to send attendance reminder *
1
2
3
4
5
6
24. It is unnecessary to send attendance reminder *
1
2
3
4
5
6
Back
Next
78%

D Sample of Mismatching Goals

Domain	Goal	User Entry	Matching goal
Nursing	Nurse responded to the call	Nurse replied to patient	Nurse talked with patient
	Nurse responded to the call	Nurse talked to the patient	Nurse talked with patient
	Nurse responded to the call	Nurse talking to person who placed call	Nurse talked with patient
	Nurse responded to the call	The nurse attended to the patient	Nurse attend to patient
	Nurse responded to the call	Nurse is in process attending to the patient	Nurse attend to patient
	Avoid nurse disturbance	Avoid making noises	Avoid patient disturbance
	Patient feels cared for	Patient is relieved	Happy patient
	Patient feels cared for	Patient is satisfied	Happy patient
	Patient feels cared for	Make sure patient is comfortable	Happy patient
	Patient feels cared for	Patient is happy and feels cared for	Happy patient
	Patient feels cared for	Patient is happy with how he is being cared for	Happy patient
	Patient feels cared for	Patient is pleased	Happy patient
Meeting Scheduler	Have meeting scheduled	Book a meeting	Book the meeting
	Book the meeting	Please schedule the meeting	Have meeting scheduled
	Book the meeting	Send meeting schedule request	Have meeting scheduled
	Book the meeting	Schedule meeting	Have meeting scheduled
	Book the meeting	schedule the meeting	Have meeting scheduled
	Book the meeting	Send meeting request	Send invitation
	Find suitable slot	Send meeting schedule request	Have meeting scheduled
	Find suitable slot	Please schedule two meetings between Monday to Wednesday	Have meeting scheduled
	Have meeting announced	Send meeting notification	Send invitation
	Request constraints by email	Please send an email	Send email
	Request constraints by email	Please have an email sent	Send email
	Request constraints by email	Please send this email via Gmail	Send email

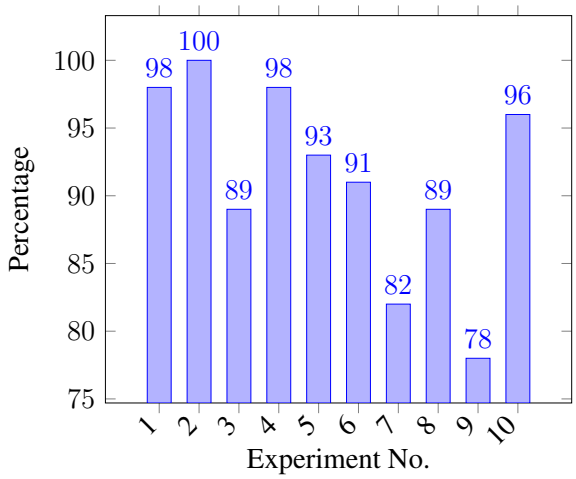
E 10-Fold Recall Analysis

Experiment No.	**Total No.**	**Successfully Split**	**Percentage**
1	67	65	97%
2	55	48	87%
3	61	49	80%
4	64	56	88%
5	28	26	93%
6	57	43	75%
7	69	51	74%
8	49	41	84%
9	32	25	78%
10	58	48	83%
Total	**540**	**452**	**84%**



F 10-Fold Precision Analysis

Experiment No.	Total No.	Correctly Splitted	Incorrectly Splitted	Percentage
1	67	65	1	98%
2	55	48	0	100%
3	61	49	6	89%
4	64	56	1	98%
5	28	26	2	93%
6	57	43	4	91%
7	69	51	11	82%
8	49	41	5	89%
9	32	25	7	78%
10	58	48	2	96%
Total	**540**	**452**	**39**	**92%**



G Task 3 - Preference Category based on Repository

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	16	94%	13%	0%	0%	0%
75%	10	70%	90%	10%	0%	0%
50%	9	0%	11%	78%	11%	0%
25%	11	0%	0%	9%	82%	9%
0%	10	0%	0%	0%	10%	90%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	14	100%	0%	0%	0%	0%
75%	15	20%	73%	7%	0%	0%
50%	7	0%	0%	71%	14%	14%
25%	13	0%	8%	38%	62%	0%
0%	14	0%	0%	0%	29%	71%

Task 3 - Preference Category based on Repository (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	12	92%	8%	0%	0%	0%
75%	10	10%	80%	10%	0%	0%
50%	5	0%	20%	80%	0%	0%
25%	6	0%	0%	33%	50%	17%
0%	9	0%	0%	0%	56%	44%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	9	100%	11%	0%	0%	0%
75%	7	14%	86%	0%	0%	0%
50%	11	9%	36%	64%	18%	0%
25%	9	0%	0%	11%	100%	0%
0%	11	9%	0%	0%	0%	91%

H Task 3 - Preference Category based on Semantic Similarity

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	7	29%	57%	14%	0%	0%
75%	13	23%	54%	8%	0%	15%
50%	5	0%	40%	20%	20%	20%
25%	6	17%	0%	0%	17%	67%
0%	7	0%	0%	0%	14%	86%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	10	40%	30%	10%	10%	10%
75%	6	17%	33%	33%	0%	17%
50%	16	19%	13%	25%	13%	31%
25%	8	0%	0%	13%	50%	38%
0%	10	10%	10%	0%	20%	60%

Task 3 - Preference Category based on Semantic Similarity (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	7	0%	29%	29%	0%	43%
75%	9	11%	22%	33%	0%	33%
50%	13	8%	8%	46%	8%	31%
25%	9	0%	11%	11%	56%	22%
0%	11	9%	0%	27%	9%	55%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
100%	8	38%	38%	13%	13%	0%
75%	6	0%	50%	17%	33%	0%
50%	7	14%	29%	29%	29%	0%
25%	9	0%	11%	22%	33%	33%
0%	5	0%	0%	20%	0%	80%

I Task 4 - Preference Category based on Repository

First Fold						
	Total	**Critical**	**High importance**	**Medium importance**	**Low importance**	**No importance**
Critical	17	47%	53%	0%	0%	0%
High importance	18	22%	78%	0%	0%	0%
Medium importance	15	0%	20%	80%	0%	0%
Low importance	13	0%	0%	31%	77%	8%
No importance	12	0%	0%	0%	25%	100%

Second Fold						
	Total	**Critical**	**High importance**	**Medium importance**	**Low importance**	**No importance**
Critical	16	50%	50%	0%	0%	0%
High importance	15	20%	73%	7%	0%	0%
Medium importance	11	0%	0%	73%	36%	0%
Low importance	11	0%	0%	0%	82%	18%
No importance	9	0%	0%	0%	11%	89%

J Task 4 - Preference Category based on Semantic Similarity

First Fold						
	Total	**Critical**	**High importance**	**Medium importance**	**Low importance**	**No importance**
Critical	21	5%	57%	10%	10%	19%
High importance	24	25%	0%	0%	75%	0%
Medium importance	23	0%	0%	100%	0%	0%
Low importance	29	0%	0%	0%	100%	0%
No importance	26	0%	0%	0%	100%	0%

Second Fold						
	Total	**Critical**	**High importance**	**Medium importance**	**Low importance**	**No importance**
Critical	22	14%	32%	23%	0%	32%
High importance	22	9%	64%	0%	23%	5%
Medium importance	21	0%	33%	10%	10%	48%
Low importance	12	0%	8%	0%	25%	67%
No importance	17	12%	6%	0%	12%	71%

K Task 4 - Detect Preference Category based on Task 2 Repository

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	10	90%	40%	0%	0%	0%
High importance	11	73%	45%	0%	0%	0%
Medium importance	4	0%	25%	75%	0%	0%
Low importance	13	0%	0%	15%	85%	0%
No importance	18	0%	0%	0%	11%	89%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	14	100%	14%	0%	0%	0%
High importance	11	45%	73%	0%	0%	0%
Medium importance	12	0%	17%	42%	25%	17%
Low importance	3	0%	0%	33%	33%	33%
No importance	10	0%	0%	10%	30%	60%

Task 4 - Detect Preference Category based on Task 2 Repository (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	11	100%	9%	0%	0%	0%
High importance	10	70%	60%	0%	0%	0%
Medium importance	8	0%	38%	63%	0%	0%
Low importance	6	0%	0%	0%	100%	0%
No importance	5	0%	0%	0%	40%	60%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	17	88%	18%	0%	0%	0%
High importance	17	47%	59%	0%	0%	0%
Medium importance	8	0%	13%	88%	0%	0%
Low importance	11	0%	0%	18%	91%	0%
No importance	10	0%	0%	10%	20%	70%

L Task 4 - Detect Preference Category based on Task 2 Semantic Similarity

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	5	60%	40%	0%	0%	0%
High importance	9	33%	67%	0%	0%	0%
Medium importance	14	14%	79%	7%	0%	0%
Low importance	7	0%	43%	0%	43%	14%
no importance	2	0%	0%	0%	0%	100%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	8	13%	63%	0%	25%	0%
High importance	10	30%	60%	0%	10%	0%
Medium importance	7	0%	57%	0%	14%	29%
Low importance	9	0%	11%	22%	33%	33%
No importance	5	20%	0%	0%	60%	20%

Task 4 - Detect Preference Category based on Task 2 Semantic Similarity (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	6	33%	33%	17%	17%	0%
High importance	2	50%	0%	50%	0%	0%
Medium importance	4	50%	25%	25%	0%	0%
Low importance	11	0%	0%	45%	9%	45%
No importance	5	0%	0%	20%	40%	40%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Critical	8	43%	0%	14%	14%	38%
High importance	6	50%	0%	33%	0%	17%
Medium importance	12	0%	42%	42%	0%	17%
Low importance	9	0%	11%	11%	22%	56%
No importance	9	22%	11%	11%	0%	56%

M Task 5 - Detect Preference Category based on Repository

First Fold					
	Total	**Absolutely Essential**	**Important**	**Would be nice**	**Unnecessary**
Absolutely Essential	22	91%	9%	0%	0%
Important	13	23%	77%	0%	0%
Would be nice	11	0%	0%	91%	9%
Unnecessary	20	0%	0%	10%	90%

Second Fold					
	Total	**Absolutely Essential**	**Important**	**Would be nice**	**Unnecessary**
Absolutely Essential	14	100%	7%	0%	0%
Important	14	7%	93%	0%	0%
Would be nice	13	0%	0%	92%	8%
Unnecessary	12	0%	0%	8%	92%

N Task 5 - Detect Preference Category based on Semantic Similarity

First Fold					
	Total	**Absolutely Essential**	**Important**	**Would be nice**	**Unnecessary**
Absolutely Essential	23	57%	26%	0%	17%
Important	22	55%	23%	5%	18%
Would be nice	23	9%	9%	61%	22%
Unnecessary	16	0%	6%	6%	88%

Second Fold					
	Total	**Absolutely Essential**	**Important**	**Would be nice**	**Unnecessary**
Absolutely Essential	23	57%	22%	17%	4%
Important	21	52%	33%	5%	10%
Would be nice	17	18%	18%	47%	18%
Unnecessary	22	18%	18%	0%	64%

O Task 5 - Detect Preference Category based on Task 2 Repository

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	11	100%	0%	0%	0%	0%
Important	12	33%	67%	8%	0%	0%
Would be nice	3	0%	0%	67%	67%	0%
Unnecessary	11	0%	0%	0%	18%	82%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	16	100%	0%	0%	0%	0%
Important	10	60%	40%	0%	0%	0%
Would be nice	6	0%	0%	67%	67%	0%
Unnecessary	17	0%	6%	0%	18%	76%

Task 5 - Detect Preference Category based on Task 2 Repository (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	13	92%	15%	0%	0%	0%
Important	9	33%	67%	0%	0%	0%
Would be nice	3	0%	33%	67%	33%	0%
Unnecessary	12	0%	0%	0%	33%	67%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	9	100%	0%	0%	0%	0%
Important	11	27%	73%	0%	0%	0%
Would be nice	5	0%	20%	80%	0%	0%
Unnecessary	9	0%	0%	0%	0%	100%

P Task 5 - Detect Preference Category based on Task 2 Semantic Similarity

Car Manufacturer						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	7	29%	57%	14%	0%	0%
Important	5	20%	80%	0%	0%	0%
Would be nice	13	0%	31%	46%	23%	0%
Unnecessary	9	0%	0%	0%	11%	89%

Meeting Scheduler						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	7	43%	0%	0%	43%	14%
Important	8	25%	38%	13%	0%	25%
Would be nice	13	0%	15%	62%	8%	15%
Unnecessary	2	0%	0%	0%	0%	100%

Task 5 - Detect Preference Category based on Task 2 Semantic Similarity (continued)

Nursing						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	8	88%	0%	13%	0%	0%
Important	7	14%	14%	29%	14%	29%
Would be nice	9	0%	33%	44%	11%	11%
Unnecessary	4	25%	25%	0%	0%	50%

Transportation						
	Total	**100%**	**75%**	**50%**	**25%**	**0%**
Absolutely essential	12	58%	33%	8%	0%	0%
Important	9	33%	56%	11%	0%	0%
Would be nice	13	23%	8%	46%	15%	8%
Unnecessary	6	17%	0%	17%	0%	67%