Goal and Preference Identification through Natural Language

Fatima Alabdulkareem
Department of Electrical Engineering and Computer Science
York University
Toronto, ON, Canada
faa@yorku.ca

Nick Cercone
Department of Electrical Engineering and Computer Science
York University
Toronto, ON, Canada
ncercone@yorku.ca

Sotirios Liaskos
School of Information Technology
York University
Toronto, ON, Canada
liaskos@yorku.ca

***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 reasoning 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.**

***Index Terms*—requirements engineering, goal modeling, natural language, decision analysis, preference analysis**

I. Introduction

Goal models [1], [2] have been widely believed to be an effective approach for modeling and reasoning about stakeholder intentions during various stages of requirements engineering. 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 [3], [4]. 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. Formal preference specification techniques have been proposed for capturing such priorities in a way that allows automated search and identification of suitable alternatives [5], [6]. 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 [4] – or later in the lifecycle when a system requires reconfiguration to meet changing needs and situations in a requirements-driven way [7], [8].

However, formal preference specification 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. These issues may hinder the effort to develop usable goal-oriented decision exploration tools that can be used by non-technical users.

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 [9], 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.

In the evaluation we conducted, 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 rephrasings 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 be effective for identifying preferential strength.

The paper is organized as follows. In Section II we offer an introduction to goal models and the natural language processing techniques that we adopt. In Section III we describe how our system is designed. In Section IV we present the design and results of our evaluation. In Section V we discuss related work and we conclude in Section VI.

II. Background

A. Goal Models and Preferences

Goal models have been extensively studied in the context of decision making during early requirements engineering [5], [6], [4] or later in the lifecycle, such as at configuration time [7], [8]. One such model of the type we consider here can be seen in Figure 1. It represents a (simplified for our demonstration here) goal model of a meeting organizer in the meeting scheduling domain. 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, soft-goals (cloud-shaped elements in the figure), which 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.

To formalize such a specification and automate the search for suitable alternatives, preference and priority specification and analysis techniques have been proposed [5], [6]. 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 preference formulations have been proposed, 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 selected goals.

Returning to our example, in a specific meeting scheduling scenario, the stakeholder may express the statements *"holding the meeting as quickly as possible is crucial"* and *"it is OK if we do not use on-line calendars"*. These expressions need processing before they can be useful for automatic reasoning. Thus, preference specification requires us to first 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., AI planner based ones [5], [6]) 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.

Fig. 1. A Goal Model

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.

B. 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 section we offer an overview of the technologies we adopt to perform the above tasks.

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 [10] and web semantics [11].

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. For example, as widely known, the character "*" matches 0 or more consecutive repetitions of any characters and "?" matches 0 or 1 repetitions. To match between either sets (e.g., a or b) "a|b" is used. Moreover, there are special characters that follow "\" such as "\s" which matches any white space, and "\W" which matches any non-alphabetic and non-numeric character. In our application we adopt Python's regular expression module and use many of the available meta-characters, particularly ? . * + | ( ) \W \s.

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 [12], such as co-occurrence methods utilizing "bags of words" [13] or more refined descriptive feature-based methods [14], [15], and corpus-based techniques, such as Latent Semantic Analysis (LSA) [16] and Hyperspace Analogues to Language (HAL) [17], which rely on information in large *corpora* (i.e., large collections of authoritative texts).

We adopt the technique used in the UMBC Semantic Text Similarity service (UMBC STS) developed by Han et al. [9], which combines LSA with WordNet [18]. The UMBC STS service adopts LSA by constructing a word-by-word co-occurrence matrix based on the analysis of a large corpus. The matrix is constructed by sliding a window of $\pm$N words, 1 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 [19], e.g., "car" and "driver" or "wife" and "marry", high-values in the co-occurrence matrix imply relatedness. Furthermore, as co-occurrence itself seems to be insensitive to alternative senses (meanings) of words, the UMBC engine utilizes WordNet [18], 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).

Given two input sentences the UMBC engine uses a complex algorithm that 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.

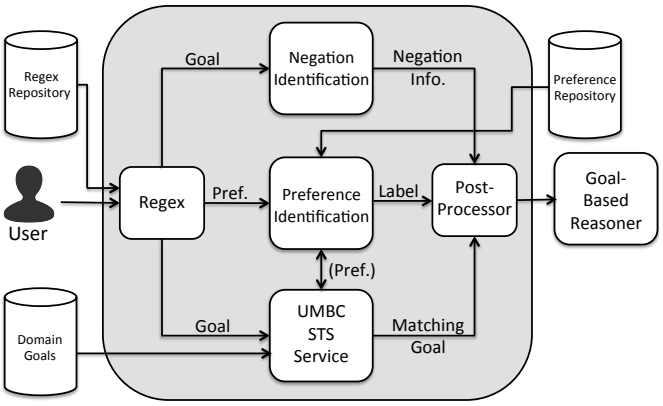

Fig. 2. System Architecture

III. An NLP System for Expressing Preferences

A. Overview

The proposed system combines the technologies we discussed in order to allow translation of preferences expressed in natural language into formal representations. Figure 2 shows the architecture of the system. 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 and preprocessed as we describe below, 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 post-processor, which constructs a formal preference. We describe these in more details below.

B. Regex

The Regex module splits the natural expressions through looking-up and applying a repository of regular expressions, the *Regex repository*. In our current proposal, 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* $\{$*schedule quickly*$\}_{Goal}$*"*), a goal followed by a preference expression (*"*$\{$*scheduling quickly*$\}_{Goal}$ *is of*

$\{$*high importance*$\}_{Pref.}$*"*), or just a goal without a preference (*"schedule quickly"*).

To see an example of the splitting process consider the sentence "*it is OK if they use on-line calendars*". In this sentence the preference part is "*OK*", while the goal part "*they use on-line calendars*". The regular expression to split this sentence: "is (.*?) if (.*)". Another example is the sentence "*I'm interested in having the meeting scheduled quickly*" the preference part is "*interested*" and the goal part is "*having the meeting scheduled quickly*". The regular expression to split this sentence is "i[\W]?(am|m) (.*?) in (.*)". The Regex will match the placeholders (.*?) and (.*) with the preference and goal parts, respectively.

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 regular expression. 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 manually ordered from specific to general. Furthermore, the system will start by checking rules that contain a preference followed 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.

C. 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. We use 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 include: *"absolutely required"* with label 100%, *"your second priority"* with label 75%, *"medium priority"* with label 50%, *"low importance"* with label 25%, and *"we rather don't need"* with label 0%.

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. If a matching key-phrase is found, then the strength label associated with the key-phrase is adopted as the preferential strength label implied in the original natural expression. If an exact match is not found, we execute the UMBC STS service between the unknown preference part and the collection of all key-phrases under each of the five labels. The label of the set 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 labels 100%, 75%, etc. (five tests) and find that it is semantically similar with the key-phrases under label 0%.

D. Building Preference Repositories from Crowd Data

Preference repositories can be developed through crowd-sourcing. 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 and all others 0. Thus a phrase such as *"scheduling a meeting is very important"* is assigned preferential strength label 100%. Alternatively, one can interpret the labels 100%, 75% etc. as samples from a continuous scale and produce a weighted average such as $100 \cdot (12/20) + 75 \cdot (8/20) = 90\%$. In our application and evaluation we followed the simple majority rule.

E. Goal Identification through 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 section. 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.

F. 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", ignoring that "little attendance" contradicts maximizing attendance.

To tackle this negation issue we populate, wherever applicable, the soft-goals of the list of goals with soft-goals 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 (as well as synonyms) of the original goal are possible. In this way, if the user refers to the negation of a soft-goal, 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 sense direct 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 hard-goals, but can also exist in expressions of soft-goals.

The above two sorts of negation information, i.e., whether the matched goal is a shadow goal and whether the natural expression is in an explicit negative form, are combined by the post-processor, as we demonstrate below.

G. 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 of the form `goal[preferential strength label]`.

As an end-to-end example, consider the natural preference expression *"it is quite desirable if the secretary works more"*, provided by the stakeholder – who, for the sake of the example, thinks secretaries do not 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".

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 through semantic similarity; label 50%. If the keyword were *"would be desirable"* instead, the exact entry would have been found in the repository with 50%.

Finally, the Negation Identification module does not detect any negation in the goal-part. Thus, the formal preference will have a negation due to the fact that a shadow goal is considered. If the goal part had a direct negation (*"... the secretary does not work more"*), the two negations, one from invoking a shadow goal and one from the detection of *'not'* in the goal part would cancel each other out.

Thus, the post-processor has all the information to construct a formal preference such as `(¬ Reduce Labor)[0.5]`; where 0.5 is the label 50%. Considering the goal model of Figure 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.

IV. Evaluation

A. Overview

As a preliminary evaluation of 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.
3) Assess the scalability and convergence of Regex, i.e., whether subsequent increments of the number of regular expressions in the Regex repository improve accuracy to a decreasing amount.
4) Explore whether a mapping from 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. Below we present the experimental design and the metrics we use to evaluate the above objectives.

B. Experimental Design

Thirty participants, twenty-four (24) male and six (6) female, are recruited from one undergraduate (16) and one graduate (14) course of the last author at York University. Their ages range from 18 to 59 years. Seventeen (17) of them

are between 21-29 years. Twelve (12) of the participants are native speakers of English.

The experiment is an on-line instrument, requiring participants to perform a series of three (3) 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 [6] (23 goals), meeting scheduler [20] (24 goals), car manufacturing [21] (32 goals), and transportation [22] (82 goals). Subjects are distributed to goal models – and therefore domains – in a between-subjects fashion: each participant is randomly assigned to one of the four domains/models. Of the four goal models, the first two have been developed by one of the authors in the past and the last two are taken from the literature.

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 step is to create a context in which the following tasks are to be understood. Participants perform then the three tasks, as we describe below.

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 in the domain. In addition, examples of goals and rephrasings are provided from other unrelated domains (driving, doing laundry, etc.).

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. Five (5) predefined levels of importance are given to the participants to consider: 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%.

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%.

C. Results

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

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. Thus, Table I shows with how many of the responses the original goal had (i) the highest similarity measure (column 3), (ii) one of the top three similarity measures (column 4) and (iii) one of the top five similarity measures (column 5).

TABLE I
MATCHING GOAL RESULTS

Domain	Responses	1st Match	Top 3	Top 5
Nursing	138	95 (68.84%)	108 (78.26%)	112 (81.16%)
Car Manufacturer	150	107 (71.33%)	136 (90.67%)	148 (98.67%)
Meeting Scheduler	151	76 (50.33%)	107 (70.86%)	132 (87.42%)
Transportation	138	89 (64.49%)	116 (84.06%)	127 (92.03%)

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.

Cases of lower precision 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 – where we also had more phenomena of semantically similar goals.

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.

TABLE II
TASK 3 ANALYSIS

Domain	No. of Statements	Incorrect Split	Could not Split	Correct Split	Pref. in Repo
Meeting Scheduler	131	9	9	113	63
Car Manufacturer	116	10	12	94	56
Transportation	105	10	11	84	47
Nursing	124	15	18	91	42
All Domains	476	44	50	382	208

2) Regular Expressions: Effectiveness and Scalability: With regards to the Regex module, we are firstly interested in the precision of the regular expressions, i.e., how well they separate 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 this component 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 and 3.

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 Regex repository. We began with twenty six (26) regular expressions which were defined before any training process. Then, we started 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 provided us with a total of 540 expressions. We divided the expressions into five (5) almost equally sized blocks. Each block was first tested. Then, expressions in the block which fail the test (don't split) despite being legitimate expressions of preference, were used as a basis to construct new rules and enrich the regular expression repository. Then we moved on to the next block and repeated the same testing-enrichment process.

As it is clearly shown in Figure 3 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 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 are the overall precision and recall? To assess precision of the splitting process, we performed 10-fold cross validation using the results of Task 2. Each fold contains natural language statements (from Task 2) by three participants. 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%.

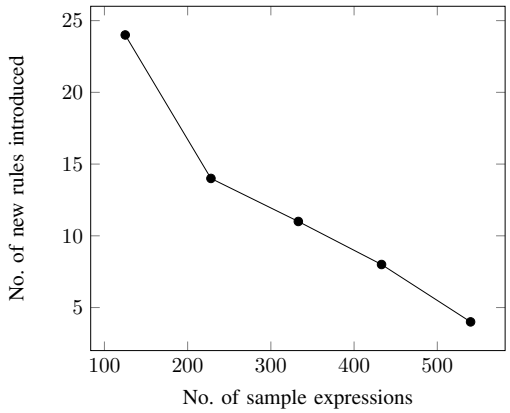

Fig. 3. New Rules Introduced vs. Examined Expression Samples

To further assess effectiveness we utilized the data from Task 3. Table II shows the results. 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 (based on manual qualitative assessment) and another fifty (50) statements could not be split by the system with the defined regular expressions, although they should have. Hence, 89.7% (382/426) of the statements that the system split were a correct split (precision), while the system was able to correctly split 80.3% (382/476) of all the statements it should have split (recall).

With respect to preference parts in the preference repository, a total of 341 preference key-phrases are identified from Task 2, 315 of which are distinct; the difference is due to classification under multiple labels. When testing with the results of Task 3, 208 of the input natural expressions lead to a pre-existing preference key-phrase, meaning that the corresponding preferential strength labels are taken directly from the repository. The remaining 174, however, were not found in the repository, meaning that for those preference parts semantic similarity is used to identify preferential strength.

3) Preferential Strength Labeling: Feasibility and Effectiveness: Our final evaluation goal asks whether a mapping between natural language expressions of preference and quantitative labels is possible (Objective 4). In particular we ask whether a crowd can help us form corpora of preference expressions from where preferential strength labels (e.g., 100%, 50% etc.) can be drawn. To evaluate this we utilize the results of Task 2 in order to develop a corpus and Task 3 to evaluate

TABLE III
INTERSECTION BETWEEN PREFERENCES IN OUR CORPUS

	Total	100%	75%	50%	25%	0%
100%	67	67 (100%)	6 (9%)	1 (1%)	0 (0%)	0 (0%)
75%	64	6 (9%)	64 (100%)	6 (9%)	2 (3%)	0 (0%)
50%	72	1 (1%)	6 (8%)	72 (100%)	9 (13%)	1 (1%)
25%	71	0 (0%)	2 (3%)	9 (13%)	71(100%)	1 (1%)
0%	67	0 (0%)	0 (0%)	1 (1%)	1 (1%)	67 (100%)

TABLE IV
DETECTING PREFERENCE CATEGORY BASED ON REPOSITORY

	Total	100%	75%	50%	25%	0%
100%	51	49 (96%)	4 (8%)	0 (0%)	0 (0%)	0 (0%)
75%	42	12 (29%)	34 (81%)	3 (7%)	0 (0%)	0 (0%)
50%	32	1 (3%)	6 (19%)	23 (72%)	4 (13%)	1 (3%)
25%	39	0 (0%)	1 (3%)	9 (23%)	29 (74%)	2 (5%)
0%	44	1 (2%)	0 (0%)	0 (0%)	10 (23%)	33 (75%)

it. 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 a collection of five sets of participant-supplied preference phrases, each set associated with a different strength. We first measure the overlap between these sets using exclusively the data from Task 2. The overlap between two sets is the number of preference phrases that are included in both sets divided by the number of preferences included in either set.

Table III 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. 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.

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, 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. Consider now the 208 of the total 382 natural expressions of Task 3 that were successfully matched with an existing key-phrase in the preference repository (which was developed using data from Task 2). In Table IV 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%". Note here that rows and columns do not necessarily add up to 100% as expressions can be classified to more than one categories, in case of a draw in support.

The same analysis was done for the 174 preferences that were not found in the repository and were, hence, matched using the semantic similarity technique. 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 V 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.

TABLE V
DETECTING PREFERENCE CATEGORY BASED ON SEMANTIC SIMILARITY

	Total	100%	75%	50%	25%	0%
100%	32	9 (28%)	12 (38%)	5 (16%)	2 (6%)	4 (13%)
75%	34	5 (15%)	14 (41%)	7 (21%)	2 (6%)	6 (18%)
50%	41	5 (12%)	7 (17%)	13 (32%)	6 (15%)	10 (24%)
25%	32	1 (3%)	2 (6%)	4 (13%)	13 (41%)	12 (38%)
0%	35	4 (11%)	1 (3%)	4 (11%)	4 (11%)	22 (63%)

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.

D. Limitations and Threats to Validity

We find the results of the empirical study to be encouraging: semantic similarity via application of the UMBC SLA-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 presence of semantically similar yet distinct goals), further experimentation 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. Thus, noting 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 options), 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 also 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. Thus, although, as we saw, this is subject for future investigation, various application contexts might be, to a certain extent, forgiving to imprecision.

V. Related Work

Natural language processing techniques have been widely used in requirements engineering for a variety of purposes. For example, Cleland-Huang et al. [23] use a classification algorithm to identify non-functional requirements within structured and unstructured texts while Weber-Jahnke and Onabajo [24] use semantic annotation ontology to analyze natural language confidentiality requirements.

Elsewhere, Yang et al. [25] use natural language techniques to detect uncertainty and speculative sentences in stakeholders natural language requirements. Comparatively, our approach differs in focus, concentrating on goals and the detection of preferences thereof from text that is intentionally specified by stakeholders given also a prepared goal model.

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 [26] and have performed studies exploring various visual aspects of goal modeling [27]. To our knowledge, the goal modeling community has not explored natural language processing for the task of evaluating goal models.

Furthermore, goal detection has been a research topic in other communities as well. Y. He, for example, [28] uses Tree-Augmented Naive Bayes network (TANs) to detect goals from natural language expressions while Casagrande et al. [29] use NLP and data mining techniques to extract goals from research abstractions and use them to create a taxonomy. Kröll et al. [30], on the other hand, propose a system to automatically annotate text with human intentions using indicative actions as a proxy for inferring such intentions. None of those efforts investigates preferences, preferential strength identification or even distinguishing preferences from goals. The technologies considered, however, could potentially be applied in our problem in the future as well.

Nevertheless, natural language expression of preferences has been studied elsewhere. Using evidence from an exploratory study involving collection of examples of participant expressed preferences, Nunes et al. [31] have proposed a meta-model for natural preference formulation which they also evaluated in terms of its usefulness for actual preference specification. In other work, they extend the meta-model to support decisions [32]. Comparatively, our work is specifically focused on goal models and reasoning therewith, aims at natural specification of preference versus a semi-structured specification approach and attempts a distinct corpus-based approach for assessing preferential strength.

Finally, the use of natural language techniques to customize preferences in configurable software systems, has also been proposed [33]. 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. While configuration is one of the application areas we aim at, in our vision such configuration is mediated by goal or other conceptual models.

VI. Conclusions and Future Work

We 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 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 our earlier proposals for preference-based software customization with an interface that could allow non-technical users perform system customizations through natural language.

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

matching, including parsing and analysis [34] of the sentence, use of probabilistic techniques [35], [28], or even common information retrieval techniques (e.g., tf-idf). Finally, we wish to extend our technique to allow detection of more expressive preference specification that our formal goal reasoning frameworks already support, such as dyadic preferences expressing orderings (as in, *"I prefer X from Y"*) and temporal constraints over goals (*"X should probably happen before Y"*). This way we will be able to deal with more complicated cases of preferences as they often occur in the real world.

Acknowledgements. This research has been made possible thanks to the generous support of King Fahad Medical City, Saudi Arabia, the Saudi Cultural Bureau in Canada, and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] F. Aydemir, P. Giorgini, J. Mylopoulos, and F. Dalpiaz, "Exploring alternative designs for sociotechnical systems," in *Proc. of the 2014 IEEE 8th International Conference on Research Challenges in Information Science (RCIS)*, May 2014, pp. 1–12.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese, "UMBC EBIQUITY-CORE: Semantic textual similarity systems," in *Proc. of the 2nd Joint Conference on Lexical and Computational Semantics*, vol. 1, 2013, pp. 44–52.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] ——, "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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] D. Heckerman and E. Horvitz, "Inferring informational goals from free-text queries: A bayesian approach," in *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, 1998, pp. 230–237.