Towards an optimal design of natural human interaction mechanisms for a service robot with ancillary way-finding capabilities in industrial environments

Cristina Aceta, Johan Kildal, Izaskun Fernández and Aitor Soroa

ABSTRACT
In addition to production-oriented robots, service robots with social skills can also perform a role in industrial environments, providing on-demand ancillary services that support production activities. In this paper, a robot that provides way-finding services within an industrial facility (e.g., finding a person or place) and is able to naturally interact with workers is presented. So as to give a more natural dimension to the robot’s verbal interaction ability and achieve acceptance from human workers, the research in this paper is directed towards the improvement of its semantic natural language interpreter, with the aim of making it able to automatically learn from new interactions. A user study is also reported, in order to assess both the capabilities of the interpreter and the performance of the robot’s communication and way-finding abilities, paying special attention to user experience (UX), which may help identify possible design problems in further research stages.

1. Introduction
Collaboration between robots and human workers is increasingly common in industrial environments (e.g., manufacturing, assembly and logistics), where they share space and production activities.

Besides production-oriented robots, service robots with a range of social skills can also play a role in production environments, providing ancillary support that may help improve work conditions, e.g., by providing location-relevant information to help reduce the perceived complexity of the environment. One such service (considered in this paper) is that the robot provides on-demand information and help with way-finding within the facilities of an organisation (e.g., finding a person, a resource, a place). Since the robot is social to some extent, a worker can communicate with it in ways that do not have to be learned (e.g., spoken natural language).

This kind of service is not process critical, and a worker may prefer to follow other conventional ways to obtain the same information such as asking another worker,

CONTACT Cristina Aceta christina.aceta@tekniker.es TEKNIKER, Basque Research and Technology Alliance (BRTA), C/ Iñaki Goeanga, 5, 20600 Eibar, Spain.
This article is an extension of the conference paper in Kildal et al. (2019).

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
trusting they may have additional tacit knowledge (e.g., where to find what). However, full connectedness of objects and people within the organisation can provide the service robot with explicit knowledge that the worker can obtain on demand, even more reliably than a fellow colleague.

So that workers choose to request services from the robot, the user experience (UX) of doing so needs to be excellent. In order to achieve it, the worker should be able to communicate naturally with the robot, as if they were interacting with a fellow worker, and the robot must optimally perform the actions it is requested to do. In this sense, the assumption is that good design decisions will lead to obtaining good UX.

The research in this work is directed towards improving the interaction between workers and robots by developing a solution that is able to learn over time, following a Lifelong Learning (Liu, 2016) approach, with special focus on the research carried out to strengthen the robot’s Natural language interpreter’s Key element extraction (KEE) module.

In this work, key element extraction resembles a NERC problem, in the sense that the key elements have to be identified and classified. The literature distinguishes between two basic approaches to NERC: rule-based and machine learning methods (Maynard et al., 2016). The use of one approach or the other depends on the circumstances of the experimentation: machine-learning-based methods are easier to maintain than rule-based ones, but require enough training data to train the models. However, despite the fact that manually generated rules are hard to maintain, rule-based systems are a more robust starting point in specific domains where there is a lack of raw or annotated data. In this sense, one of the main challenges in industrial scenarios is the lack of annotated data – specifically in languages other than English –, necessary for the training task of systems based on supervised methods. For this, an initial implementation of the semantic natural language interpreter, which includes a KEE module that relies on a hand-made set of rules that semiautomatically tag key elements in human commands, has been developed. The use of these rules enables the creation of training data to be used with supervised methods, which do allow the development of the system following an incremental learning approach.

A user study has been carried out and reported in order to evaluate both the performance of the Semantic interpreter and the robot navigation module quantitatively, reporting performance results – with a special mention to the research carried in the interpretation of commands in real time through the interpreter’s KEE module – and qualitatively, focusing on how the effect of the design of specific aspects of the robot influences user experience.

The paper is structured as follows: Section 2 presents related work to robot interaction systems and the basic architecture involved and user experience (UX). Section 3 covers the context of the investigation performed in this work. Section 4 describes the semantic natural language interpreter that will be used in the experimentation. Section 5 includes the description and qualitative and quantitative evaluation results of the user study, performed in a real scenario. Section 6 provides discussion on the results obtained in this paper.

2. Related work

As stated in the sec:intro introduction, in order to trigger acceptance from human workers towards robots it is very important that the robot’s user interface is ‘properly designed’ (Villani et al., 2018), meaning that the interaction is intuitive enough for the worker to be
able to concentrate on their current tasks rather than on the specific communication with the robot (Villani et al., 2018). Moreover, the robot should be able to properly communicate with the user in terms of system behaviour (e.g., problems, feedback). Thus, the use of techniques that require the full attention of the worker (mice, keyboards, screens) would be left aside (Villani et al., 2018).

One of the challenges that may arise in human-robot interaction in industrial settings is environmental noise, which causes automatic speech recognition (ASR) systems to reduce their performance. In order to overcome noise conditions, most approaches in the literature have adopted simpler voice-communication systems that rely on specific structures and vocabulary (Bugmann & Pires, 2005; Veiga et al., 2009). Nevertheless, limiting the type of interaction the user can have with the robot does not comply with simplifying workers’ tasks (Villani et al., 2018).

Out of the context of social robots in industrial settings, several service robots with ‘humanlike voice communication capabilities’ that are able to follow instructions given by humans and even follow a dialogue can be found in the literature, with a wide range of implementation options (M.A.V.J. Muthugala & Jayasekara, 2018). Although each implementation has different settings, the following common modules can be distinguished: (1) Automatic Speech Recognition, (2) Key element extraction from commands, (3) Storage of robot’s world knowledge, and (4) Interaction problem solving. These modules will be commented in the following lines.

The Automatic Speech Recognition component is necessary to process a voice command’s transcription in further interpretation steps. The most extended practice is to make use of ASR systems that are capable of processing free-form language (that is, without any restrictions of vocabulary or syntactic patterns). An example is Google Speech API (GoogleLLC, 2019), Cutugno et al. (2013) and Matuszek et al. (2014). Other approaches, however, depend on systems that are limited to a certain vocabulary (Puigbo et al., 2015). In the case of the latter, if an out-of-vocabulary word is uttered, the system would not accept the command and a new one would be requested.

For key element extraction – in this case, from commands –, both rule-based and data-driven proposals can be found, as stated in the previous section and in Bastianelli, Castellucci, Croce, Basili et al. (2014). According to the authors, rule-based systems are more robust; however, they require an expert to construct and maintain them. On the other hand, data-driven approaches are not that robust (although the use of annotated examples in training improves the performance (Bastianelli, Castellucci, Croce, Basili et al., 2014)) and do not require the work of an expert in the long term. As an example of a rule-based system, Puigbo et al. (2015) present a Context-Free Grammar (CFG)\(^1\) that is able to obtain, from a previously POS-parsed command, its action (mainly the verb) and references to objects, people and locations. In Zukerman et al. (2015), however, a previously syntactically parsed command is translated into conceptual graphs for its interpretation. As for data-driven approaches, Kollar et al. (2010) use a CRF-trained algorithm in order to extract four main relevant elements from each command: ‘figure’ (the subject of the sentence), a ‘verb’ (an action to take), a ‘landmark’ (an object in the environment), and a ‘spatial relation’ (a geometric relation between the landmark and the figure)\(^2\). In this line, in order to overcome the lack of annotated data in industrial scenarios,\(^2\) necessary to train the supervised models, there are several intents of generating useful data for robotic/industrial settings. An example is Human-Robot Interaction
Corpus (HuRIC) Bastianelli et al. (2014), which is an HRI dataset in English, generated from interactions with a house service robot.

Regarding the storage of the robot’s world knowledge, the literature distinguishes between different techniques to make the world accessible to the robot in order to determine the actions to be executed over which world elements and the implications of its actions on it. In Zukerman et al. (2015), this knowledge is stored in a database form, in which the state of the context (i.e., the world) is described, whereas in Muthugala and Jayasekara (2016a, 2016b) a hierarchical structure is used, in which the rooms and primary and secondary objects (e.g., kitchen – table – plate, respectively) are illustrated. Finally, the approach in Puigbo et al. (2015) is based on an ontology, in which the information about the locations (maps), people and objects are modelled. In this sense, ontologies are easy to maintain and update with new terms. Also, their reasoning and inferencing capabilities provide them with consistent knowledge and the possibility of deriving new information from existent data, which make them a very good option in human-robot settings, which need reliable and updated information for the scenario for which they are modelled.

As for the problems that may arise in the interpretation of the command, ambiguity or insufficient information are some examples. In order to solve ambiguity, Fasola and Zukerman make use of score systems so as to determine the interpretation that complies better with the model of the world. Regarding insufficient information (or an unfruitful attempt to solve ambiguity), common practice is to ask the user for clarification (Muthugala and Jayasekara 2016a), Muthugala and Jayasekara (2016b), Kollar et al. (2010), Puigbo et al. (2015), and Zukerman et al. (2015)). In addition, it is possible to use this clarification step to learn new terms and improve the system through user feedback, as in Muthugala and Jayasekara (2016b).

In the same way as with any scenario for human-robot interaction (HRI), the user experience (UX) elicited by an industrial service robot needs to be assessed (Weiss & Huber, 2016). ISO (DIS, 2010) defines UX as ‘a person’s perceptions and responses that result from the use or anticipated use of a product, system or service’. This definition opens up the opportunity for assessing UX from the earliest stages of the development of a robot. In addition to the effectiveness of the collaboration, the factors that are known to influence the UX obtained in HRI scenarios, which include trust (Hancock et al., 2011) and perception of safety (Amini et al., 2013), need to receive attention from early design stages. UX is subjective, dynamic and context-dependent (Law et al., 2009). For that reason, it has to be measured in user studies that can obtain a snapshot of this metric, which may in future serve as baseline for the analysis of the evolution of such UX, once in use (Wurhofer et al., 2015).

The research in this paper contributes to the introduction of social robots in industrial settings by designing a Semantic natural language interpreter that is able to process commands uttered in natural language, without wording or syntactic restrictions. This interpreter both benefits from rule and supervised approaches in order to minimize the number of errors in interpretation, with an added capacity of learning over time through new interactions. The system proposed in this work also makes an attempt to overcome the lack of data for industrial settings, necessary to train the supervised models, by using out-of-domain data for training. Additionally, the use of an ontology to model the world provides a well-structured means of storing and relating the elements in the scenario as
well as the actions that the robot is capable of executing. This system also allows to obtain the different possible interpretations for a possible command through inference. The combination of rules and supervised methods, the use of an ontology and the capability of the system to ask the user for confirmation reduces the number of interpretation errors, and prevents the robot from executing wrong tasks in case of erroneous interpretations. Also, since the system is able to learn from new interactions, it will eventually substantially minimize the work of an expert in order to improve its performance. Moreover, the analysis of the UX of this system implemented in a service robot in a real environment setting allows to identify the modules which require more attention in further improvement tasks.

3. Experimental context

A service robot deployed within an industrial environment to offer ancillary support should fulfil the following a priori requirements (in line with Johnson et al. (2017) and Sohn et al. (2018)):

(1) To be clearly distinguishable in its appearance from other robots and autonomous vehicles that do carry out process-critical tasks.
(2) To be able to navigate autonomously in a dynamically changing indoor environment, with moving people that may get in its way.
(3) To be able to communicate verbally with a human user, both by understanding natural speech input and by producing speech output.
(4) To have accurate and up-to-date knowledge of the industrial facility, its contents and the people working in it.

As it can be seen, these requirements are limited to aspects related to navigation (#1 and #2) and communication with the robot (#3 and #4).

As one of the key a priori requirements, communication between humans and service robots in industrial settings plays a very important role, since a good user experience in this setting generates acceptance from human workers, that may choose to rely on service robots to the same extent as fellow workers. If the human command is not understood correctly, the dialogue process is bound to fail and, eventually, no acceptance from the user will be achieved. Moreover, it deals directly with natural language, freely expressed by the worker without any formal restrictions, which sets an added complexity to the interpretation task.

So as to guarantee a natural communication between both agents, it is necessary to develop a robust semantic natural language interpreter that ensures such interaction. In this sense, the investigation in this paper in this matter will be directed towards the improvement of the interpreter.

It is possible to develop the natural language interpreter so as it is able to learn over time through supervised methods, which is a plus that allows continuous improving of the solution. However, the annotated data in Spanish for this type of task and in industrial settings in general is scarce and complicates the development process.
Finally, it is important to determine to what degree the implementation of the previously mentioned requirements influence the quality of the user experience (UX) so that new requirements could be added to the design and corrective measures could be implemented.

### 3.1. Project design

As mentioned in the previous section, this work will be focused in two main aspects: (1) the ability of a service robot designed to provide ancillary support in industrial environments to successfully communicate with human workers and (2) the capability of said robot to be able to navigate autonomously and safely to the destination it is requested to.

An existing guide robot, KTBot (Susperregi et al., 2012), was recruited as the starting point for a prototype (see image of the robot in Figure 1).

In order to cover the communication capability of the robot, the experimentation in this paper is focused on providing the robot with a more natural dimension in interaction with humans, by making it able to learn from new interactions through its semantic natural language interpreter and, more specifically, its KEE component. For this, the use of supervised methods, which allow an incremental learning approach (Liu, 2016) and are easy to maintain in the long run, is necessary. However, the lack of annotated data for industrial settings affects negatively the development of such a system. In this sense, two options arise: (1) to manually tag a corpus in order to obtain training data and (2) to develop a linguistic tool that allows to automatically tag commands directed to a robot and, thus, the creation of an annotated corpus.

Although the use of supervised methods is the final objective, manually tagging data is extremely costly. Thus, the approach in this paper is the creation of a linguistic tool for KEE that, being initially developed using hand-made rules and implementable in the semantic natural language interpreter as the KEE module, also allowed the semiautomatic creation of a corpus that could be used to train supervised models to improve or complement the KEE component.

Regarding navigation, KTBot’s embodiment offers an approachable appearance distinguishable from other production equipment. It is capable of simultaneous localisation and mapping (SLAM), including numerous sensors for obstacle detection and safe navigation.

By providing KTBot with the capacity of interaction with humans by implementing the Semantic natural language interpreter – with a rule-based version of the KEE module –, a user study has been conducted. In this study, to provide the robot with capacity to interpret a natural voice request, a mobile application that queries a semantic natural language interpreter – described in Section 4 –, consisting of several natural language processing modules, has been implemented. This user study has allowed the creation of the semiautomatically annotated corpus used to train the models necessary for a KEE module based on supervised methods, which has also been evaluated. In order to evaluate the generalization capacity of each approach, experimentation with data from a different albeit similar domain is performed.

A final evaluation step has been performed with the results in the user study in order to assess the performance of both the Semantic natural language interpreter and the whole cycle of interaction (communication + navigation) and obtain early indications
about which elements in the HRI scenario (focusing on navigation and communication with the robot) affected UX, both positively and negatively.

4. Semantic natural language interpreter

Following the requirements for a robotic system in an industrial environment pointed in Section 3, the semantic natural language interpreter must ensure a natural verbal communication between the worker and the robot and have up-to-date facility knowledge.

The semantic natural language interpreter proposed in this paper follows a modular structure (see Figure 2), and is intended to be queried by a mobile application the worker interacts with by uttering requests directed to the robot. An Automatic Speech Recognition system – commonly known as ASR –, given an uttered request in Spanish, transcribes it into text. The speech recognition is performed using Google Cloud’s Speech-To-Text API, the input of which is a recording of the command, and the output the corresponding transcription.
As depicted in Figure 2, using the transcribed command obtained by the ASR system, the flow of information is the following: first of all, a Key Element Extraction component aims to retrieve its most relevant information. Secondly, a semantic-based Semantic interpreter that relies on a Semantic Knowledge Manager makes the corresponding inferences to find the equivalent robot actions that best fit the key elements obtained in the previous step. Finally, the robot-understandable actions are sent to the robot to be executed. Each of the modules that take part in the semantic natural language interpreter are described in the sections below.

4.1. Knowledge manager

The Knowledge Manager aims to describe the relevant elements taking part in the interactions in a logic and intuitive way, in the same way, information is structured in the human brain.

For that purpose, the Knowledge Manager (KM) is based on an ontology (defined in OWL) to model the environment and the robot capabilities as well as the relationships between them, which can be understood as implicit rules that the reasoner can exploit to infer new information. It also supports the description of the scenario status: which operation is executing, if it involves a robot, if the robot is running something or waiting for the start of a program, etc.

The ontology, which has been developed reusing existing ontologies, includes three core classes in order to model the scenario knowledge:

- foaf:Person. Reused from FOAF, corresponds to people that work in the building.
- geo:SpatialThing. Reused from GEO, corresponds to rooms, floors, objects and other elements that are in the spatial extension of the building. Those elements can be classified in the following subclasses⁵:
  - Room. Reused from DERI Vocabularies.⁶
  - Object

Figure 2. Graphical description of the information flow in the interaction between the user (using a mobile application) and the robot, including the different modules that the Semantic natural language interpreter includes.
- Site. Reused from DERI Vocabularies. As stated in the ontology, ‘an area of land with a designated purpose, such as a university Campus, a housing estate, or a building site’.
- Floor. Reused from DERI Vocabularies.
- Desk. Reused from DERI Vocabularies.
- geo: Point. Reused from GEO, ‘a point, typically described using a coordinate system relative to Earth’.
- FloorSection. Reused from DERI Vocabularies. Corresponds to the different sections a floor may be divided in (e.g., the laboratories area).
- Building. Reused from DERI Vocabularies.
- geo:TemporalThing. Reused from GEO, makes reference to the actions that can be performed upon those spatial elements. The actions have the following subclasses, according to the class of the element the action has an effect on:
  - ActionOnPerson. If the action has an effect on elements categorized as Person (foaf:Person)
  - ActionOnObject. If the action has an effect on an Object (Object)
- ActionOnSpace. If the action has an effect on a Space (e.g., Room, Floor)

Figure 3 shows an excerpt of the ontology.

In order to model the relations between actions and spatial elements as well as people and their characteristics, a set of properties have been defined. These properties, depicted in Figure 4, in some cases, have an inverse property and establish restrictions on the elements that may have them. The main relations modelled are the following:

![Ontology Diagram](image-url)

**Figure 3.** Excerpt of the ontology, in which the core elements (Spatial elements, actions and people) and their subclasses are included.
4.2. **Semantic interpreter**

The aim of the Semantic Interpreter is to check the coherence and feasibility of the frames coming from the Semantic text interpreter. When more than one result is obtained, the
The coherence and feasibility checking of a given action in a certain situation is supported by the information in the Knowledge Manager. After obtaining the action and destination elements from the uttered command through the KEE tool, the Semantic interpreter consults the Knowledge Manager to check the existence of both elements in the ontology. If there is compatibility, the interpreter, through reasoning, obtains the corresponding robot-interpretable actions, ordered according to probability measures given the action and destination used for inferring. So as to make sure the user agrees with the inferred destination, the system asks the user for confirmation. If the request coming from the voice channel is compatible and all the necessary information is available, it sends the action to the robot. When the request does not become an order for the robot due to incompatibilities found in this step, the user is notified why the request has not been successfully sent to the robot (no action identified, incompatibility between the request and the situation, etc.).

4.3. Semantic text interpretation – key element extraction

Once the human request has been transcribed using the ASR module, the Key element extraction (KEE) module aims to retrieve the most relevant information from said command as key elements. These key elements, thus, contain the core information from the command.

In order to achieve a natural interaction between human workers and robots, the approach is to improve its Key Element Extraction (KEE) component by using supervised methods, so as to make it able to learn incrementally – that is, over time – from new interactions.

One of the objectives of the semantic natural language interpreter and thus, of the KEE system, is to develop it following a Lifelong Learning (LL) approach (Veron et al., 2019), in which the system is able to automatically learn from new sequences, preserving previous knowledge. This incremental learning approach favours a more natural communication, since the system would learn from new interacting formulas from the user. The use of supervised methods would allow such functionalities but, unfortunately, the annotated data for the domain to be used for training are scarce. Given the lack of domain annotated data in Spanish for the guidance scenario, necessary to develop a supervised key element extraction system, a set of rules were defined to identify and classify the key elements from commands in Spanish.

In the methodology used in this work, a set of hand-made rules are modelled and implemented in the KEE tool to semiautomatically tag a domain and an out-of-domain datasets. This semiautomatic tagging consists on the use of the rule system, implemented as a REST API, to label each command’s key elements, followed by a manual correction of the initial tagging performed by an expert. The resulting datasets will be used to train the models in order to develop a machine-learning-based KEE tool.

4.3.1. Rule-based KEE

As a first step to obtain annotated domain data, the rule-based key element extraction component from the semantic natural language interpreter, in which a set of rules were
defined to extract the key elements from a given command in the guidance domain, has been used. Note that the rules do not rely on external data but on domain knowledge, so this method overcomes the data availability problem for the domain.

The input of the system is a syntactic tree for a given command (see Figure 5), which is then matched against the rules. The motivation behind this procedure is that the key elements to be extracted follow specific syntactic structures. For instance, actions usually correspond to verbal forms, whereas destinations tend to follow nominal structures (nouns and their modifiers). Also, in order to correctly assign to a verb its corresponding destination (or vice versa), the predicates (that is, the part of the sentence that gives information about the subject, which combines the verb and its arguments) are also extracted. For example, for the command ‘Ir al aula de robótica’:

- Command: ‘Ir al aula de robótica’
- Predicates: [Ir al aula de robótica]_{pred}
- Key elements for each predicate: [[Ir]_{verb} al [aula de robótica]_{verbArgument}]_{pred}

The rules were described using Foma (Hulden, 2009), which is an open-source tool that allows to define the key element structures and the rules for their identification and classification, and manually written by an expert by inspecting the examples included in the experimentation presented in (Susperregi et al., 2012) and, thus, they can be

![Figure 5. Syntactic tree for the sentence ‘Quiero café.’ (‘I want coffee.’). For this sentence, the picture corresponds to the graphic representation of the tree, whereas the tree below the image would be the input for the rule-based tool. Each number in the input matches the index of the word the tree element makes reference to.](image)
considered as expressing the domain knowledge obtained by an expert by introspection of the data.

In order to create the rules, the first step is to define the structures that the elements to detect follow. Example 1 shows the definition of an element and the observations that justify it.

(1) Definition: DEST elements are Nouns + their modifiers (prepositional phrases, numbers, adjectives, etc.).
   (a) Quiero café. (Single Noun) - I want coffee.
   (b) Quiero una sala de reunión. (Noun + Prepositional phrase ('de reunión')) - I want a meeting room.
   (c) Llévame a la sala 3. (Noun + Number ('3')) - Take me to room 3.

After defining the elements, the rule that tags them is created. Examples 2 and 3 show a definition and a rule as they would appear in the grammar, respectively.

(2) Definition: 'A Dest element can be a single noun (SNDest), or can be followed by an Adjective Phrase (SAdj), a Number (SNNum), or a Prepositional Phrase (SPDest)'.
   (a) define Dest [SNDest | SAdj | SNNum | SPDest]*

(3) Rule: 'Once a Dest element, which has been described previously, has been detected, place a tag at the beginning and at the end of the sequence. Also, the longest match will be selected (@->)'.
   (a) define TagDest [Dest @-> "<DEST>" ... "</DEST>" ];

Therefore, given an input tree and a set of definitions and rules, the key element extraction system returns the same tree with each key element delimited by tags. For the sentence in Figure 5 ('Quiero café.'), the input and output from the rule-based system are shown in Example 4:

(4) a. Input: (grup-verb:1(verb:1)(sn:2(grup-nom-ms:2(n-ms:2)))(F-term:3))
   b. Output: (grup-verb:1<GV-DEST>(verb:1)</GV-DEST><DEST>(sn:2(grup-nom-ms:2(n-ms:2)))</DEST>)
   c. Output elements:
      • Verb: (verb:1) → Quiero
      • Dest: (sn:2(grup-nom-ms:2(n-ms:2))) → café

If two rules match the same tree sequence, the longest match is selected. Complex sentence structures (such as coordination and subordination) are not currently fully supported, although present in the dataset, since they may be supported in the future.

This approach for KEE has been implemented in an initial version of the semantic natural language interpreter and tested and evaluated in a real-environment experimentation, described in Section 5.1.
4.3.2. ML-based KEE

So as to obtain a KEE component that is able to learn over time following a Lifelong Learning approach (Liu, 2016) through supervised methods, it is necessary to have access to annotated data to train the models. In order to do so, domain and out-of-domain data have been used to develop the system. The use of out-of-domain data to augment existent domain training data has been proved to obtain positive results in the literature (Fromreide & Søgaard, 2014; Persson, 2017).

The following lines will describe the Domain and Out-of-Domain data, along with the features and algorithms used to develop the model.

**Domain data.** IMH is a dataset in Spanish for the guidance scenario. It consists of a set of commands directed to a guide robot, obtained through the experimentation task described in Section 5.1. The dataset was semiautomatically annotated using the rule-based KEE tool presented in Section 4.3.1, manually revised and corrected when necessary.

In total, IMH includes 179 sentences, with a total of 1151 words and 1232 tokens and an average of 6.43 words per sentence. The actions and destinations expressed in the sentences are tagged, as these are the key elements to be identified and linked to the ontology. The action is the element that describes the act to perform, associated with the destination (‘drink water’), whereas the destination is the spatial point where the user wants the robot to lead them. It can designate a specific point (‘meeting room’) or an element related to the destination point (‘coffee’ for ‘coffee machine’).

All in all, the IMH corpus is annotated with the following tags:

- **ACTION.** *I want to go to a meeting room with a PC.*
- **DEST-TARGET.** Target element. *I want to go to a meeting room with a PC.*
- **DEST-PREP.** Containing preposition in a prepositional phrase. *I want to go to a meeting room with a PC.*
- **DEST-COMPL.** Element that depends on the previous preposition. It conveys the element that is contained in a target element. *I want to go to a meeting room with a PC.*

Following usual practice, the corpus at the word level is annotated using the BIO schema that marks each word as beginning, being inside or being outside of a sequence, respectively. Table 1 shows an example of a sentence with the corresponding tags.

**Out-of-domain data.** The out-of-domain corpus used in the linguistic tool experimentation is HuRIC (Bastianelli, Castellucci, Croce, Iocchi et al., 2014). This dataset contains English sentences from a domain close to guidance – the domain in IMH –, though different, since it consists of interactions with house service robots in three different tasks. HuRIC comprises 318 sentences, with 2618 words and 2697 tokens and an average of 8.21 words per sentence. The commands in HuRIC are tagged taking into account linguistic information, such as morphosyntactic and semantic information. However, the tags initially included with the corpus are not used for this experimentation, but the same tags as in IMH.
So as to increase the size of IMH using HuRIC, the latter was formatted in order to be compatible with the first. Firstly, HuRIC was automatically translated to Spanish using Google Translate services, without performing any revision, and then the rule-based system was applied over it, in order to obtain the same tag set as IMH. Then, the tags assigned by the rule-based system were used to train the supervised systems. To assess the effect of introducing noisy data in the training, a second version where the classification errors were manually corrected was created. This task was performed by a linguist, and the time required to review each sentence depended on three factors:

- The quality of the automatic translation. If the translation is difficult to understand or ungrammatical, the expert may require more time to mentally determine the key elements.

(5) a. *Puede lograr que el taza para el sofá de la sala de estar por favor.* - **Bad translation quality**
   *Can you achieve that the cup for the living room sofa please.*
   b. Buscar la llave en el baño. – **Good translation quality**
   Search for the key in the restroom

- The number of correct tags. The more correct tags the sentence has, the less time required to correct. Similarly, the more incorrect tags, the more time required.
- The length of the command. The shorter the sentence, the less time required to review.

For a good quality translation, the time for the expert to detect the key elements approximately went from 5 to 10 seconds, whereas for a bad quality translation the time invested could increase up to 45 seconds. Regarding the correction task, it would take from 0 to 30 seconds, depending on the number of incorrect tags. Higher timings also occurred on lengthier commands.

Table 2 shows the size of IMH and HuRIC – both hand-reviewed and automatically generated versions – datasets, including the annotated elements.

**Features.** The features used to perform supervised analysis are the syntactic features extracted from the syntactic tree described in Section 4.3.1 along with the cluster each token belongs to according to a set of pre-computed Clark Clusters for Spanish (Agerri & Rigau, 2018). Clark clusters allow to deal with word variations or synonyms that express...
similar meaning, and which usually fall into the same cluster. The features of the previous and next token are also considered to represent the current token. To sum up, the following features are considered:

- Related to tree structure: *terminal*, *parent1*, *parent2*, and *parent3*. *terminal* stands for the syntactic element in the lowest position in the tree. *parentN* is the N-th parent of a non-terminal category in the syntactic tree counting from lower to higher position.
- Other relevant features: *cluster*. The cluster the token belongs to.

Table 3 includes the features obtained for a given example sentence. In this approach, the features related to tree structure are ordered from the highest to the lowest position in the tree, and the marks for Gender (masculine/feminine) and Number (singular/plural) have been removed to provide a generic analysis.

**Algorithms.** The following algorithms have been considered to develop the tool:

- Support Vector Machine (SVM). A classifier is trained to assign a single tag to each word. The possible tags are described in Section 4.3.2.
- Conditional Random Fields (CRF). In this setting, key extraction is considered as a sequence labelling problem. The format of the tags to predict is the same as in SVM.

Linear SVM has been used as provided by the Weka tool (Hall et al., 2009) and, for the CRF method, the python-crfsuite library implementation. Default parameters have been used in all the methods, as no fine-tuning was performed.

**5. Evaluation**

This Section covers all the evaluation performed in the scope of this work. The evaluation results will be presented from more specific to more general modules. First of all, both approaches to the KEE component (using rule-based and supervised methods) will be
analyzed, taking into account their capacity to extract key elements from domain data and their behaviour when data from a different but similar domain are provided.

Also, the performance of the whole Semantic natural language interpreter will be reported, focusing on the modules of the pipeline and their contribution to a correct or incorrect interpretation of a given command.

Finally, the complete system (communication and navigation) behaviour in the user study performed will be reported quantitatively (providing figures on performance) and qualitatively (providing UX results).

5.1. Experimental setup

In order to evaluate the performance of the Semantic interpreter – using the rule-based KEE presented in previous sections – and the navigation capabilities of the service robot, a user study has been conducted. Furthermore, the rule-based KEE module allows the capability of semiautomatically annotating the commands in the study in order to obtain an annotated domain corpus to be used to develop a KEE module based in supervised methods.

As stated in Section 4.1, the ontology exploited by the Semantic interpreter is instantiated for this use case and available as an RDF repository through Stardog. This allows to have an endpoint to easily access and infer information from the ontology using the standard SPARQL query language for RDF. This endpoint is the connection point between the Knowledge manager and the Semantic interpreter module.

In this study, the users interacted with a guide robot in a large research centre and, after a series of spoken interactions, intended by the semantic interpreter to confirm or disambiguate the requests received, the robot navigated to the destination. For this, users were introduced to the task by being provided information about the environment the robot was placed in – through a map of the facility – and the goal of the experimentation: to ask the robot to guide them to a designated destination of their choice from the aforementioned map.

Requests could be explicit (e.g., ‘take me to the auditorium’) or implicit (e.g., ‘I’m tired’, for which going to the sofa could be an adequate response). However, in order to ensure a natural interaction between the users and the robot and not influence their requests, the participants were not instructed about possible structures that could be directed towards the system.

The experimentation is described in the following sections. For the study, 29 participants – 3 females – were recruited, with an age range between 21 and 52. Twenty-six of these participants – two females – were aged 21 to 25.

Twenty-seven of the participants – age range 21 to 28, 2 females – were final-year students from a dual engineering degree that combines academic training with practical experience in manufacturing industries, whereas the remaining two – aged 40 and 52, 1 female – were instructors at that institution.

This profile provided the study with participants that had an up-to-date awareness of the new manufacturing technologies, first-hand experience in processes at manufacturing companies, plus an open-mind towards new technologies, a characteristic found in young prospective industry professionals (Kildal et al., 2018). However, the participants did not have any experience working with robots or naturally interacting with them.
Although the discussion in this paper revolves around the use of social robots in industrial settings, the experimentation carried out was performed at the lobby area of a research centre facility. The reasons to perform the experimentation outside an industrial environment are the following:

- The available robot navigation capability. The robot had a map of the facility hall implemented for navigation but not of other areas, such as the workshops in the research centre Tekniker.
- Security. The user study included 29 people, which were considered to be a great number of people to be in a real industrial setting in order to comply with security regulations (e.g., distance between users and machines, PPEs [Personal Protection Equipment]).
- Although the study has been conducted in the facility hall, previous research with the same ASR system – Google Speech API – as in this work has been conducted at Tekniker’s industrial laboratory environment, with encouraging results.
- Regarding User Experience, we considered that the environment chosen would allow a sufficient number of users to interact with the robot in the real setting (vid. Security).

The study was structured in two parts, as detailed below.

### 5.1.1. Voice interface and interaction cycle

The complete interaction cycle had three parts: (i) the participant made a natural voice petition to the robot (using an Android-based mobile device with the voice interpretation application installed); (ii) the robot proposed (in natural speech) the best-fit interpretation to check with the participant if it was correct. If not, another interpretation was proposed until one of them was accepted; and (iii) the robot started to navigate towards the destination at a speed of 0.7 m/s.

The robot negotiated obstacles (e.g., a person) that got in its way (by stopping, waiting and looking for an alternative route if necessary). Once the destination was reached, the robot announced it verbally. If the destination was not reachable for the robot (e.g., because of a staircase), the robot provided detailed spoken directions to the participant.

The goal of this first session was to assess the voice-based natural interaction cycle. All 29 participants were divided into three groups of 9–10 members. Each group gathered around a table in a meeting room, with two experimenters managing each session. The mobile device with the voice interpretation application installed was circulated around the table. Participants were asked to imagine that they were at the lobby area of the building with the service robot (they had earlier visited the lobby area and had seen the robot), emulating what would be a real interaction with the robot. Each participant was handed a copy of the lobby floorplan shown in Figure 6. The plan included lists of destinations that they could request to be guided to by the robot: specific labs or workshop areas, people by their name (a list of 10 names included in the knowledge repository was shown) plus meeting rooms with their actual names and other destinations with feature descriptions. Real industrial environments might have other features described, but the mechanics of interaction would be the same. The session lasted 40 minutes. Each participant completed 3–4 interaction cycles (without physically navigating to any
destination) and they also observed the interactions of all other participants (30–40 per session).

To assess the UX provided by the voice interaction cycle, each participant filled in an ad-hoc 12-questions 5-point Likert scale questionnaire. Each question was a statement, and the scale rated the degree to which the participant agreed with it. Statements were phrased in positive or negative terms regarding six aspects of the voice interface and interaction cycle: (i) ease of use, (ii) naturalness of speech with which the system could be addressed for it to work, (iii) the appropriateness of the options returned by the system in response to a petition, (iv) latency in responding, (v) how well the system understood petitions, and (vi) the agility of the interaction. For each one of these six aspects about the system, there were two statements in the questionnaire (in shuffled order), one of which made a positive statement about that aspect and the other one a negative statement, opposite in meaning to the corresponding positive statement. However, the wording used in both statements was different, so that it was not immediately evident that each question appeared twice with its polarity inverted. This is a methodological technique that is used to strengthen the internal consistency of the responses, and to identify participants that might be responding randomly, in which case responses of such participant should be discarded (Anderson et al., 1983). This questionnaire configuration and analysis procedure, which has also been seen in other works such as Sauro and Lewis (2011), corresponds to the standard SUS questionnaire (Brooke, 2013).

5.1.2. Complete way-finding service through interaction with the robot
Eleven of the 29 participants (one female, all students) also participated in the evaluation of the complete way-finding service provided by the robot (including petition and navigation). The session was individual for each participant. It took place in the lobby
area with the robot and two experimenters that facilitated the session. Other people external to the study occasionally crossed this same space.

Each participant completed three full interaction cycles (from petition to reaching destination), starting each navigation where the previous one had ended. After each cycle, the participant rated on two 7-point Likert scales (i) how easy it had been to complete the cycle (i.e., the validated SEQ (Sauro, 2012) questionnaire), and (ii) the satisfaction with the result obtained. At the end of the session, the participant completed two post-study questionnaires. First, an ad-hoc questionnaire was used to evaluate the UX of interacting with and navigating after the physical robot. Each of the eight questions was a statement, and the response rated the degree to which the participant agreed with it, on a 5-point Likert scale. Statements were phrased in positive or negative terms (see methodological clarification at the end of Section 5.1.1) regarding four aspects of the interaction with the physical robot: (i) situation awareness, (ii) speed of movement of the robot, (iii) perception of safety and (iv) reliability of the service. After that, a validated System Usability Scale (SUS) that referred to the complete system and service (voice petition with the mobile application and navigation to the destination) was requested to be filled. Finally, comments and opinions about the interactive experience were collected in a face-to-face interview.

5.2. **Key element extraction tool evaluation**

This section evaluates the performance of both rule and supervised approaches to develop a linguistic tool for KEE in Spanish. First, results are reported by using domain data, both for training and testing. A second experimentation was also performed by adding data from a different yet similar domain to the existent domain data for training and then testing in out-of-domain data, in order to determine the capability of the systems to generalize or deal with unseen interactions. This last setting resembles the case of an incremental learning system that is initially trained with domain data but has to deal on production time with new out-of-domain examples.

Domain data were obtained from the experimentation described in Section 5.1 and out-of-domain data from the experimentation in Bastianelli et al. (2014). Both datasets are described in 4.3.2.

5.2.1. **Evaluation using domain data**

The experiments performed in order to evaluate using the rule-based and supervised methods used the IMH dataset as domain data.

In order to perform the experiments, the dataset was split into ⅔ for training – for supervised methods – and ⅓ for testing. For evaluation, following usual practice, standard precision, recall and F1 measures used on sequence labelling tasks are reported, which were calculated over whole chunks, using the CoNLL18 evaluation script for the NER task. Additionally, the values for accuracy at a word level are also reported.

Table 4 shows the results of the methods. The rule-based method obtained good results, but the supervised systems yielded better results in general. Overall, the CRF method performed best with an F1 of 91.4% and an accuracy of 96.34%. Nevertheless, it is important to keep in mind that the size of the test dataset is very small. As a consequence, an error in classification would drastically decrease the evaluation measures.
Table 4. Results of $\frac{2}{3}$ train – $\frac{1}{3}$ test for key element identification and classification algorithms on IMH corpus.

| Algorithm | Precision | Recall | F1     | Acc  |
|-----------|-----------|--------|--------|------|
| Rules     | 83.92     | 88.89  | 86.33  | 95.30|
| SVM       | 82.43     | 90.37  | 86.22  | 94.78|
| CRF       | 92.42     | 90.37  | 91.39  | 96.34|

Table 5. Results for key element identification and classification algorithms on IMH corpus. Values for accuracy.

| Algorithm | Accuracy |
|-----------|----------|
| Rules     | 96.79    |
| SVM       | 84.01    |
| CRF       | 96.66    |

Table 5 shows the results of evaluating the methods over the whole IMH, using 10-fold cross-validation techniques for the supervised methods and traditional accuracy calculation for rules, due to the difficulties of performing cross-validation on data obtained in such circumstances. In this case, the evaluation is slightly different and only accuracy at a word level is reported. In this case the rule-based system was the system that performed better, with an accuracy of 96.79%. Nevertheless, the difference between rules and CRF is insignificant: 96.79% and 96.66%, respectively.

5.2.2. Evaluation using domain and out-of-domain data
The rule-based and supervised linguistic tools were also evaluated using domain along with out-of-domain data in order to determine the generalization capacity of both approaches when new examples (from a similar but different domain) appear.

The aim of the experimentation performed in this section was to resemble a domain shift scenario. So as to assess the impact of adding out-of-domain data – and correcting it or not – to the domain training set, the systems described in Sections 4.3.1 and 4.3.2 were applied. For the test set, a $\frac{2}{3}$ of HuRIC was used (hand corrected). For the training set, the following configurations were considered:

- Domain data, using the whole IMH for training.
- Domain data and out-of-domain data. In this setting, two phenomena were analysed: the effect of augmenting the domain data with out-of-domain data and the effect of correcting the latter or not, in order to evaluate whether the automatic tagging of the data is enough or it is necessary to hand-review out-of-domain data for this task. Thus, the following training set configurations were assessed:
  - IMH plus the raw (not corrected) $\frac{2}{3}$ fraction of HuRIC.
  - IMH plus the hand corrected $\frac{3}{3}$ fraction of HuRIC.

Table 6 shows the results of the experiments, adding also the figures for the rule-based system. Overall, the rule-based system performed worse than the supervised systems regardless of the training set used (82.03% F1 vs 85.87% from the ML
algorithm with the lowest F1). This shows that hand curated rules are difficult to adapt to new domains, and that supervised systems generalize better, favouring an incremental supervised approach.

However, the table also shows that noisy data hurt the performance of the supervised systems (86.28% F1 for the best algorithm with non-corrected out-of-domain data vs 98.09% F1 for the same algorithm, but correcting out-of-domain data). In fact, in this setting, it was better to use a smaller – but correct – training set than to augment it with noisy data (91.16% vs 86.28% for the best algorithm – CRF – in the two settings). The best results were obtained when using the hand-corrected version of HuRIC, including information from the testing domain (98.09% F1 for the best algorithm).

All in all, taking into account the results obtained, supervised methods will be considered in a future update to the KEE system, either to be combined with rules or to replace them.

5.2.3. Error analysis

As an additional step, the chunks obtained with the rule-based system were compared with the results obtained by the best system in Table 6, CRF with corrected out-of-domain data. The goal was to analyse which structures could not be resolved by any of the systems, and the type of structures that the supervised system was able to solve that the rule-based tool is not, and vice-versa.

There are 34 chunks which were not correctly identified and classified by either model. By analysing those 34 chunks, it was noted that the rule-based system only performed better in detecting certain destinations formed by single elements of the category noun (e.g., sofa, table, bed). On the other hand, CRF was able to solve a certain complex structure that was not correctly detected by the rules. This structure is the following:

(6) \(X \text{ \{Adjective\}} \text{ de \{Determiner\} }Y\), being \(X\) and \(Y\) words that belong to the category noun.

(a) ‘Sofá de la habitación’ – ‘Sofa of the room’\(^{15}\)
(b) ‘Extremo de esta tabla’ – ‘End of this table’
(c) ‘Lado derecho de la cama’ – ‘Right side of the bed’
Although it seems unlikely to be able to justify the effect of one structure when the difference between the baseline and CRF (hand-rev) is of a 16% in F1 (82.03% vs 98.09%), it is important to point out that this structure represents a 17.64% (more specifically, 6 cases) of the 34 total chunks revised. That is, it is a fairly common structure in the dataset. Furthermore, CRF also successfully tagged certain verbs that the baseline did not, and these verbs also represent an important part of the total chunks: a 50% (17 cases).

The chunks that could not be identified and classified by either model are 13. These chunks included, for example, sentences with a complex structure and elements that were not correctly tagged by the syntactic parser, as it can be seen in [ex:allwrong-errors] Examples 7 and 8, respectively. Example 7a provides a case of recursive prepositional phrases (a prepositional phrase inside another) and 7b a command that contains a relative clause. In Example 8a, estar was analyzed as a verb (since isolated is indeed a verb) and not as a part of a nominal whole, sala de estar.

(7) Complex structures
   (a) [caja de la izquierda de la grabadora]
      [box at the right of the tape recorder]
   (b) [cojin negro que se encuentra en la cama]
      [black pillow that is on the bed]

(8) Not correctly parsed
   (a) [sala de estar]
      [living room]

5.3. Semantic natural language interpreter evaluation

The performance of the totality of the semantic natural language interpreter has been evaluated from analysing the results obtained in the experimentation. For this, the commands have been classified according to three situations:

- Correct. The interpreter has been able to extract the intended robot-translated command and, thus, is sent to the robot.
- Incorrect. The command has not been interpreted correctly and either the robot is sent a different robot-translated command than the intended or is not sent any command at all. The error is the result of a bad analysis by any of the modules in the interpreter.
- Invalid. Invalid commands are commands generated due to external problems (ambiental noise interpreted as a command, misuse of the application, etc.).

Table 7 includes the results of the command classification of the 156 commands in the experimentation. As it can be seen, more than half of the commands (54.49%) have been correctly interpreted, and the number of invalid ones is marginal (9.62%).
In order to determine the source of the errors, the incorrect interpretations have been analyzed in more detail, taking into account the results obtained by each of the modules that take part in the interpretation.

The results in Table 8 show, out of the Incorrect commands, the distribution of the errors for each module:

- **KEE.** The KEE module has not been able to extract the correct keywords or predicates (as in the example below).

  (9) a. Command: ‘Me gustaría ir al aula de robótica’ (I would like to go to the robotics class)
  b. Keywords: {gustar\textsubscript{verb}}\textsubscript{pred1} \{ir\textsubscript{verb}; aula de robótica\textsubscript{target}\}\textsubscript{pred2}
  c. Keywords extracted by the system: \{gustar\textsubscript{verb}\} \textsubscript{pred1} \{ir\textsubscript{verb}\}\textsubscript{pred2}
  \{aula de robótica\textsubscript{target}\}\textsubscript{pred3}
  d. Error: The system has divided \textsubscript{pred2} in two separate predicates, tearing apart the verb \textit{(ir)} and its corresponding argument \textit{(aula de robótica)}.

- **Semantic interpreter/Ontology.** The keyword is not present in the ontology or the ontology has not been able to infer the destination correctly.

  (10) a. Command: ‘Llévame al retrete’ (Take me to the loo)
  b. Error: The destination that represents \textit{retrete} (the toilet) is modelled in the ontology, but not associated with the terminology \textit{retrete}, but with other terms such as \textit{baño} or \textit{aseo}.

- **Other**
  - **ASR.** The ASR system has not transcribed the command correctly.
    (11) a. Original command: Llévame a la sala Ura 1
    b. ASR command: ‘llévame a la sala 11’
  - **Parser.** The parser has not analyzed the command correctly.

  (12) a. Command: Quiero ver a Asier Cuevas.
    b. Correct analysis: Quiero/VB ver/VB a/PREP asier/NP cuevas/NP.
c. Parser’s analysis: Quiero/VB ver/VB a/PREP asier/ADV cuevas/NP

- Out-of-scope requests. The user has requested an action that the robot cannot perform.

(13) a. Command: ‘llame a emergencias’ (Call Emergency)
  b. Error: The robot cannot make calls.

In this sense, an error comes from a module when the wrong interpretation has its origin on said module.

In the scope of this work, in which the focus has been put in the KEE module and the semantic approach to interpretation, the attention has to be directed to the KEE (i.e. key element extraction errors) and Semantic Interpreter/Ontology (i.e. reasoning errors) columns, and especially to the former, since part of the research in this work has been focused on its improvement. Both modules represent a 10.71% of errors in the interpreter each. Although the evaluation results show that around a 21% of the errors are caused by both modules combined, in a real setting the consequences of a wrong interpretation would cause a somewhat minimal impact on the actions of the robot itself, in the sense that the robot would not be sent any command to execute without the user’s confirmation, as described in Section 4.2.

According to the results obtained in Section 5.2 regarding the capacity of using supervised methods for the KEE tool and, thus, learning over time, the results for the KEE tool can be improved consistently using supervised methods (combining with rules or substituting them). The improvement of the Semantic interpreter is beyond the reach of this work, though considered as a line of research in the future. This line of research will be focused on the capacity to automatically include out-of-vocabulary terms in the ontology through user feedback in order to minimize errors, along with a revision on the structure of the ontology so as to make inferencing more robust. In order to obtain user feedback, the implementation of dialogue capabilities will be explored.

As a side note, the modules in the Other category (except Out-of-scope requests) are third-party tools and, in this sense, future work may include testing other options that suit better the conditions of the system’s use case.

All in all, the results obtained through the implementation of the rule-based semantic natural language interpreter in a real-life experimentation – and gathered in this section – suggest that there is still room for substantial improvements in the system.

5.4. Complete system evaluation

Though the previous sections have dealt with the evaluation of specific modules in the interaction cycle, it is also important to have a general vision of the complete system. For this, quantitative and qualitative analyses have been performed. Regarding the quantitative analysis, the results in the interpretation and navigation modules are put in perspective. For the qualitative analysis, the UX of the whole process of interaction is evaluated.

5.4.1. Quantitative analysis: navigation to final destination

As the Semantic natural language interpreter has already been evaluated in Section 5.3, it is important to determine which of the commands that have been interpreted correctly (and thus, were translated to robot-computable orders and sent to the robot) completed the interaction cycle successfully by correctly navigating to the desired destination.
Table 9 provides the percentage of successful and unsuccessful interaction cycles given a correct interpretation. The results obtained in this sense are promising, since the robot was able to successfully navigate to a 70.58% of the destinations it was requested to. Most of the unsuccessful navigations were caused by obstacles (e.g., people) and the inability of the robot to resume the navigation to the final destination.

5.4.2. Qualitative analysis: user experience evaluation

As specified in Section 5.1, the experimentation is structured in two parts. For this, the UX evaluation has been performed taking into account the obtained information from each of the tasks.

The analysis of data collected from the first experimental session (voice recognition interaction, 29 respondents) is summarised in Figure 7. In this graph, there is a box for each category analysed, which combines participants’ adherence to the positive and negative statement about the same category (29 scores in the 0 to 4 range and another 29 in the 0 to −4 range make up each box). For every category, results show that the adherence to the positive judgement (e.g., ‘The app was easy to use’) was statistically significantly higher to the adherence to the negative statement. According to this, participants thought (left to right) that (i) the application was easy to use, (ii) participants found that they could speak to it naturally, (iii) it returned options that were seen as appropriate, (iv) response time was acceptably low, (v) it understood well the petitions, and (vi) the interaction flowed in an agile way.

Figure 8 shows a similar analysis of the data from the UX assessment of the interaction (navigation) with the robot. Right to left on the graph, it shows statistically significantly that (iv) the service was thought to be reliable and (iii) the robot was perceived to be safe. However, and also statistically significantly, (ii) the speed of the movement of the robot was thought to be too slow. The only result without a statistically significant difference

|                | Correct navigation | Incorrect navigation |
|----------------|--------------------|----------------------|
| Correct interpretation | 70,58%             | 29,41%               |
between positive and negative judgements was Situation awareness \[ t(10) = 2.23; p = 0.52 \]. The corresponding questions were phrased in terms of whether the participant could tell why the robot sometimes interrupted its movement, which happened when it encountered an obstacle in its way. However, this was not always clear for the observing participant.

The analysis of the SUS questionnaire revealed a SUS score of 80.68, which corresponds to good usability of grade ‘A-’ (Sauro, 2011). Individual questions received high average scores (top quartile of the scale). As the exception, answers to the question ‘I think that I would like to use the system frequently’, which captures the overall subjective impact, averaged above the middle of the scale. The after-task SEQ question (scale 1 to 7) revealed that the task was consistently found to be easy to carry out \( M = 6.55 \ SD = 0.87 \). In contrast, the satisfaction question on the same scale obtained only average scores and a much broader dispersion \( M = 5, SD = 2.32 \). The post-study interviews revealed that sources of dissatisfaction with the result obtained originated from events and system behaviour aspects such as the robot stopping without apparent reason, poor assumptions made by the natural voice interpreter, or an incomplete construct of the world in the semantic database.

### 6. Conclusion and future work

The research on this paper has been directed towards a social robotic system that provides ancillary support in industrial environments. These robots are not process-critical and the use of the robotic service basically depends on the UX obtained, which should be excellent in order to trigger acceptance from workers. The main sources of this acceptance – and, thus, are subject to a UX analysis – is that the communication between human workers and the robot feels natural, as if the user was talking to a fellow worker, and that the robot is capable of performing the actions it is requested to do optimally.

In this work, the research has been oriented to a natural human-robot communication in Spanish through a Semantic natural language interpreter and, especially, its KEE component. The aim has been to follow a Lifelong learning approach (Liu, 2016) in its development, that would allow learning over time and, thus, a continuous automatic improving of the solution. In order to do so, the use of supervised methods is necessary. However, the data necessary to train the models for industrial scenarios...
in Spanish and for this application is scarce. In order to overcome this, a rule-based KEE has been designed. Besides being implementable in the Semantic language interpreter as KEE component, the rule-based system also allows the obtention of annotated data that, eventually, will allow the use of supervised methods. In order to assess the performance and UX of both the communication between users and the robot – by implementing the rule-based KEE in the Semantic natural language interpreter – and the navigation capabilities of the robot, a user study has been conducted. Furthermore, this study also has allowed to obtain an annotated domain dataset in Spanish and, thus, the first implementation of a KEE component based on supervised methods.

The evaluation process in this work has been reported from more specific to more general modules. Regarding the KEE component, both the rule- and the supervised-based methods have been evaluated using domain data – obtained in the experimentation – and, then, adding data from a similar although different domains in order to determine their capacity of generalising when new structures appear. Although rules obtain very good results in domain data, supervised methods based on CRF perform better (86.33% vs 91.39% in F1).

Out-of-domain experiments have also been carried out by training on a combination of both domain and out-of-domain datasets and testing on the out-of-domain dataset, HuRIC (Bastianelli, Castellucci, Croce, Iocchi et al., 2014), which was automatically tagged with rules – and also a hand-corrected version was created. The results in this setting show a significant drop in performance of the rule system, which stresses the difficulty such systems have to adapt to new (albeit related) domains. The supervised systems perform best, with an error reduction of up to 89% compared to the rule-based system. However, a considerable drop is observed when training them with automatically tagged data (and, therefore, not hand-corrected). This suggests that both SVM and CRF are sensitive to noisy data so that both algorithms perform better using fewer but less noisy data. Thus, supervised methods can automatically learn from new sequences – a process that in a rule-based system would require a lot of manual effort – and can contribute to develop a more robust Semantic natural language interpreter through its KEE component.

The evaluation on the Semantic natural language interpreter in the experimental task shows that more than half of the commands have been interpreted correctly by the system. Also, the impact of each of the modules in the interpreter in bad interpretations is also assessed, and both the KEE component and the ontology represent a 10.71% of the errors in the interpreter each. Taking into account the results obtained for the KEE component, it is possible to substantially improve these results by implementing supervised methods for KEE. Regarding the improvement of the information stored in the ontology, future research will be centred in the capacity of automatically adding new terms via user feedback, which could be obtained through the implementation of dialogue capabilities. Future work will also include the exploration of other third-party tools for ASR and syntactic parsing in order to increase the ratio of correct interpretations of the system, as 28.57% of the errors of the system were originated in these two modules.

Finally, the performance of the system as a whole has been evaluated quantitatively and qualitatively. The quantitative performance showed that, given a correct interpretation by
the interpreter, the robot was able to correctly navigate in a 70.58% of the cases, which is a good result. Most of the problems in navigation were caused by obstacles in the way of the robot, which caused it to totally stop.

The qualitative evaluation has been performed in terms of UX. Although the system as a whole scored a high usability grade (A-), in its current form it failed to inspire participants to want to use it frequently.

Regarding human-robot communication, the natural interaction with the system was rated very positively. Although other individual characteristics of the system were rated highly, limitations of the description in the semantic repository were identified as affecting UX negatively. Moreover, the speed of the robot and situation awareness were also reported as negative features in terms of UX.

All the areas mentioned previously require corrective measures for the system and the service to provide a better UX. In the short future, these three main issues will be approached as described below.

The speed of the robot is easy to increase. However, safety regulations are likely not to permit speeds that would have satisfied participants in this study. Having to follow a robot is bound to be frustrating unless the user is walking at a normal pedestrian speed. Anything below that may appear inefficient, particularly when the service should be supporting workers' productivity.

The solution to this as further work could be a redesign of the service itself. However, as a short-term solution, a good strategy would be that the robot outlined some paths to follow prior to navigating. Situation awareness is a source of uncertainty, which is known to affect UX negatively expectations (Lohse, 2009). This problem seemed to be most acute when the robot stopped (for no clear reason to some). Techniques such as displaying robot status and intentions as well as outlining the pathways (e.g., through simple but effective multimodal display designs) will be applied.

Lastly, regarding the Semantic natural language interpreter, participants found that the application made naïve assumptions and simplifications. The ontology will be enriched to have a broader facility description, and modifications in the interpreter will be performed to provide more context when showing the best interpretations to the user.

Future work considers a new evaluation and validation of the whole system in a real industrial environment, in the context of production activities. This new experimentation will include the actions outlined previously and the implementation of the KEE component based on supervised methods, in order to determine the impact of incremental learning in interpretation on both the quality of the interpreter and the user experience, along with the improvements mentioned.

Notes

1. In a nutshell, CFGs are formal notations that account for regular patterns in a language.
2. *vid.* Introduction.
3. See Maurtua et al. (2017) for a similar architecture.
4. https://cloud.google.com/speech-to-text
5. Only the subclasses that may cause confusion to the reader are defined in more detail.
6. http://vocab.deri.ie/; http://vocab.deri.ie/rooms
7. http://www.w3.org/2003/01/geo/wgs84/pos
8. https://protege.stanford.edu/
References

Agerri, R., & Rigau, G. (2018). Language independent sequence labelling for opinion target extraction. Artificial Intelligence, 268, 85–95. https://doi.org/10.1016/j.artint.2018.12.002

Amini, P., Schmitt, R., & Bregulla, M. (2013). Perception of safety. ATZ Worldwide, 115(5), 50–56. https://doi.org/10.1007/s38311-013-0062-2

Anderson, A. B., Basilevsky, A., & Hum, D. P. (1983). Measurement: Theory and techniques. In P. H. Rossi, J. D. Wright & A. B. Anderson (Eds), Handbook of Survey Research (pp. 231–287). Elsevier.

Bastianelli, E., Castellucci, G., Croce, D., Basili, R., & Nardi, D. (2014). Effective and robust natural language understanding for human-robot interaction. In T. Schaub, G. Friedrich & B. O’Sullivan (Eds.), ECAI’14 - Proceedings of the Twenty-first European Conference on Artificial Intelligence (pp. 57–62). IOS Press.

Bastianelli, E., Castellucci, G., Croce, D., Iocchi, L., Basili, R., & Nardi, D. (2014, May). HuRIC: A human robot interaction corpus. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14) (pp. 4519–4526). European Languages Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/pdf/531/Paper.pdf

Brooke, J. (2013). SUS: A retrospective. Journal of Usability Studies, 8(2), 29–40. https://dl.acm.org/doi/10.5555/2817912.2817913

Bugmann, G., & Pires, J. N. (2005). Robot-by-voice: Experiments on commanding an industrial robot using the human voice. Industrial Robot 32(6), 505-511.https://doi.org/10.1108/01439910510629244

Cutugno, F., Finzi, A., Fiore, M., Leone, E., & Rossi, S. (2013). Interacting with robots via speech and gestures, an integrated architecture. In F. Bimbot, C. Cerisara, C. Fougéron, G. Gravier, L. Lamel, F. Pellegrino & P. Perrier (Eds.), INTERSPEECH-2013 (pp. 3727–3731). ISCA.
DIS, I. (2010). 9241-210: 2010. Ergonomics of human system interaction - Part 210: Human-centred design for interactive systems. In International Standardization Organization (ISO). Switzerland.

Fromreide, H., & Søgaard, A. (2014). NER in tweets using bagging and a small crowdsourced dataset. In A. Przepiórkowski & M. O gordniczuk (Eds.), Advances in Natural Language Processing (pp. 45–51). Springer International Publishing.

Google LLC. (2019). Cloud speech-to-text. Retrieved February 20, 2020 from https://cloud.google.com/speech-to-text/

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter, 11(1), 10–18. https://doi.org/10.1145/1656274.1656278

Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y., De Visser, E. J., & Parasuraman, R. (2011). A meta-analysis of factors affecting trust in human–robot interaction. Human Factors: The Journal of the Human Factors and Ergonomics Society, 53(5), 517–527. https://doi.org/10.1177/0018720811417254

Hulden, M. (2009). Foma: A finite-state compiler library. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session (pp. 29–32). Athens, Greece.

Johnson, M. J., Johnson, M. A., Sefcik, J. S., Cacchione, P. Z., Mucchiani, C., Lau, T., & Yim, M. (2017). Task and design requirements for an affordable mobile service robot for elder care in an all-inclusive care for elders assisted-living setting. International Journal of Social Robotics, 12, 989–1008. https://doi.org/10.1007/s12369-017-0436-5

Kildal, J., Fernández, I., Lluvia, I., Lázaro, I., Aceta, C., Vidal, N., & Susperregi, L. (2019). Evaluating the UX obtained from a service robot that provides ancillary way-finding support in an industrial environment. In Y. Jin & M. Price (Eds.), Advances in Manufacturing Technology XXXIII: Proceedings of the 17th International Conference on Manufacturing Research, incorporating the 34th National Conference on Manufacturing Research (pp. 61–66). Queen’s University, IOS Press.

Kildal, J., Tellaeche, A., Fernández, I., & Maurtua, I. (2018). Potential users’ key concerns and expectations for the adoption of cobots. Procedia CIRP, 72, 21–26. https://doi.org/10.1016/j.procir.2018.03.104

Kollar, T., Tellex, S., Roy, D., & Roy, N. (2010). Toward understanding natural language directions. 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (pp. 259–266). Osaka, Japan.

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P., & Kort, J. (2009). Understanding, scaling and defining user experience: A survey approach. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 719–728). Boston, USA.

Chen, Z. & Liu, B. (2016). Lifelong Machine Learning. Morgan & Claypool Publishers.

Lohse, M. (2009). The role of expectations inHRI. In K. Dautenhahn & J. Saunders (Eds.), New Frontiers in Human-Robot Interaction (pp. 35–56). John Benjamins Publishing Company.

Matuszek, C., Bo, L., Zetlemoyer, L., & Fox, D. (2014). Learning from unscripted deictic gesture and language for human–robot interactions. In C. Brodley & P. Stone (Eds.), AAAI14: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (pp. 2556-2563). AAAI Press.

Maurtua, I., Fernández, I., Tellaeche, A., Kildal, J., Susperregi, L., Ibariguren, A., & Sierra, B. (2017). Natural multimodal communication for human–robot collaboration. International Journal of Advanced Robotic Systems, 14(4), 4. https://doi.org/10.1177/1729881417716043

Maynard, D., Bontcheva, K., & Augenstein, I. (2016). Natural language processing for the semantic web. Synthesis Lectures on the Semantic Web: Theory and Technology, 6(2), 1–194. https://doi.org/10.2200/S00741ED1V01Y201611WEB015

Muthugala, M. A. V. J., & Jayasekara, A. G. B. P. (2018). A review of service robots coping with uncertain information in natural language instructions. IEEE Access, 6, 12913–12928. https://doi.org/10.1109/ACCESS.2018.2808369
Muthugala, M. V. J., & Jayasekara, A. B. P. (2016a). Enhancing human–robot interaction by interpreting uncertain information in navigational commands based on experience and environment. 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2915–2921). Stockholm, Sweden.

Muthugala, M. V. J., & Jayasekara, A. B. P. (2016b). Mirob: An intelligent service robot that learns from interactive discussions while handling uncertain information in user instructions. 2016 Moratuwa Engineering Research Conference (MERCon) (pp. 397–402). Moratuwa, Sri Lanka.

Persson, A. (2017, May). The effect of excluding out of domain training data from supervised named-entity recognition. In Proceedings of the 21st Nordic Conference on Computational Linguistics (pp. 289–292). Association for Computational Linguistics.

Puigbo, J.-Y., Pumarola, A., Angulo, C., & Tellez, R. (2015). Using a cognitive architecture for general purpose service robot control. Connection Science, 27(2), 105–117. https://doi.org/10.1080/09540091.2014.968093

Sauro, J. (2011). A practical guide to the system usability scale: Background, Benchmarks & Best Practices.CreateSpace Independent Publishing Platform.

Sauro, J. (2012). Measuringu: 10 things to know about the single ease question (seq). Retrieved March 28, 2019 from https://measuringu.com/seq10/

Sauro, J., & Lewis, J. R. (2011). When designing usability questionnaires, does it hurt to be positive? Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (p. 2215–2224). Association for Computing Machinery.

Sohn, K., Morris, E., Ulebor, O., Currier, T., & Merrill, S. (2018). Service robot design for uses in human centered environments. In Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition. Volume 4A: Dynamics, Vibration, and Control. Pittsburgh, Pennsylvania. ASME. https://doi.org/10.1115/IMECE2018-86479.

Susperregi, L., Fernández, I., Fernández, A., Fernández, S., Maurtua, I., & López de Vallejo, I. (2012). Ubiquitous computing and ambient intelligence. In J. Bravo, D. López-de Ipiña, & F. Moya (Eds.), Interacting with a robot: A guide robot understanding natural language instructions (pp. 185–192). Springer Berlin Heidelberg.

Veiga, G., Pires, J., & Nilsson, K. (2009). Experiments with service-oriented architectures for industrial robotic cells programming. Robotics and Computer-integrated Manufacturing, 25 (4–5), 746–755. https://doi.org/10.1016/j.rcim.2008.09.001

Veron, M., Ghannay, S., Ligozat, A.-L., & Rosset, S. (2019, April). Lifelong learning and task-oriented dialogue system: What does it mean? In International Workshop on Spoken Dialogue Systems Technology. Siracusa.

Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics, 55, 248–266. https://doi.org/10.1016/j.mechatronics.2018.02.009.

Weiss, A., & Huber, A. (2016). User experience of a smart factory robot: Assembly line workers demand adaptive robots. In M. Salem, A. Weiss, P. Baxter & K. Dautenhahn (Eds.), Proceedings of the 5th International Symposium on New Frontiers in Human-Robot Interaction 2016. Sheffield, UK. arXiv preprint arXiv:1606.03846. https://arxiv.org/abs/1606.03846.

Wurhofer, D., Meneweger, T., Fuchsberger, V., & Tscheligi, M. (2015). Deploying robots in a production environment: A study on temporal transitions of workers’ experiences. IFIP Conference on Human-Computer Interaction (pp. 203–220). Bamberg, Germany.

Zukerman, I., Kim, S. N., Kleinbauer, T., & Moshtaghi, M. (2015). Employing distance-based semantics to interpret spoken referring expressions. Computer Speech & Language, 34(1), 154–185. https://doi.org/10.1016/j.csl.2015.01.002