Proceedings of the International Workshop on Intelligent Information Systems
IIS'2011, September 13-14, 2011, Chisinau, Republic of Moldova

Adding a Question Answering Component to a Robot Using Public Ontologies

Tudor-Alexandru Dobrilă, Adrian Iftene

Abstract: Robotics is a very broad and wide domain, which is rapidly evolving. In recent years, researchers have tried to simplify the process of software development for robots, which culminated with the launch of Microsoft Robotics in 2008. This paper shows a software system built using this technology and tested in the real world, on a real robot. Our goal was to create components that simulate aspects of human behavior and to give it artificial intelligence. One of the most important components that we added to our robot and that is presented in this paper is the question answering component.

Keywords: Robotics, Question Answering, Semantic Web.

1 Introduction

The authors for IIS'2011 Workshop Proceedings are requested to follow instruction given in this sample paper. This template provides authors with most of needed formatting specifications.

Bill Gates, the founder of Microsoft, declared in an interview for the magazine "Scientific American" (January 2007 edition), that he predicts a future in which robots will be "part of our daily lives", as a consequence to the latest breakthroughs in distributed computing, wireless communications and image and sound analysis. Existing platforms in the field of robotics, such as Toddler[1], allow the implementation of several operations on robots: *movement, object tracking*, etc.

The system proposed in this paper is organized in a Service Oriented Architecture (SOA) manner. Services were designed for *robot movement, communication* with the peripheral devices (sensors, webcam, microphone, etc.), *memory management* (in XML format) and interaction with humans.

The robot's memory contains all the information required to execute algorithms for face detection, face recognition, edge detection, image segmentation and classification and shape similarity (Embedded Hidden Markov Models for face detection, an image database, etc.). The question answering (QA) component implements the conversion of ontology in OWL format to an AIML file [3], which is stored in the robot's memory

[1] Toddler: http://www.parallax.com/dl/docs/books/edu/toddler.pdf

and used together with an open-source implementation of AliceBot[2] to answer questions from humans. The memory can be automatically extended, by searching ontologies on the Internet and converting them.

2 QA Component

In order to develop a robot capable of answering questions, we have approached a method proposed by Eric Freese in [1] of converting an ontology (semantic web, in OWL format) to an AIML file [3] (a pattern-based format specified using XML). The system was implemented in Java and it is capable of generating about 15.000 patterns in 2 minutes from an ontology (in OWL format) containing roughly 1.000 entities. The main advantage of this approach is the small reply time and flexibility, granted by the use of constructions similar to regular expressions, which can contain wildcards ("*" and "_"). In this way 15.000 patterns cover millions of different formulations.

In our application, in order to extend the knowledge base of the robot, we added the capability to query a semantic web search engine, Swoogle[3] (by accessing Swoogle's web service in a RESTful manner). The ontologies returned can then be converted to AIML format and saved in the robot's memory. This way the learning process is significantly speeded up. The conversion is preceded by a preprocessing phase, in which the ontology is extended with definitions from a lexical database (such as WordNet) and with information from Wikipedia. Data from Wikipedia can be collected by sending HTTP requests to Wiki's API[4] and is done in three steps: 1) Keyword-based search for finding relevant pages; 2) Extracting content, in wiki format, from a page; 3) Parsing text in wiki format and converting it to plain text.

In order to design a language independent system, we use XML files, which contain templates that will be later used to create AIML categories. These templates can contain wildcards ("*" or "_") and several keywords ([*name*], [*value*], etc.), which are replaced at run-time by values from the ontology. With these language files at hand, the OWL to AIML conversion is done by querying the ontology using Protege API[5], an open-

[2] AliceBot: http://www.alicebot.org/
[3] Swoogle: http://swoogle.umbc.edu/
[4] Wikipedia API: http://en.wikipedia.org/w/api.php
[5] Protege API: http://protege.stanford.edu/plugins/owl/api/

source library written in Java, and one of the many reasoners available (for instance Pellet[6]).

This process is based on the concept on question domain from [2]. Questions can be classified in the following categories, based on the answer that is expected:

• **Definition questions** (e.g. "*What is * X*?"): the superclasses of the *X* class and its restrictions are retrieved from the ontology and are used to create an answer. For instance, for a class named **queue**, which belongs to **data structure** and **collection** and has a restriction **keeps_elements_in** *some* **order**, a conversation could be of the form: Q: "*What is a queue*?" A: "*A queue is a type of data structure and collection.*"

• **Measure questions** (e.g. "*How many types * X **?"): the list of subclasses of *X* is retrieved from the ontology and is used to build an answer. For instance, for a class named **variable_order**, a conversation could be of the form: Q: "*How many types of variable orders do you know*?" A: "*I know two types of variable orders: dynamic variable order and static variable order*."

• List **questions** (e.g. "*What types of X **?"): similar to the measure-type questions.

• **Comparison questions** (e.g. "_ *X * Y*?"): the super classes and restrictions of both X and Y are retrieved and are used to search for similarities/differences between the two classes.

• **Factual questions** ("*Which X fulfills the Y property*?"): the subclasses and restrictions of *X* are retrieved. For each restriction *Y*, an AIML category is generated. For instance, for a class **problem**, which has a subclass **constraint_satisfaction_problem** with a restriction **is_solved_by** *some* **inference**, a conversation could be of the form: Q: "*What types of problems are solved by inferences*?" A: "*Types of problems that are solved by inferences: constraint satisfaction problems*."

We wanted to generate general patterns and several tools were needed to make grammatical changes to words: **Lemmatization**: MorphAdorner[7], an open-source library written in Java; **Verb conjugation**: After finding the root form of the verb we used Verbix[8], an online tool, to conjugate it;

[6] Pellet reasoned: http://clarkparsia.com/pellet
[7] MorphAdorner: http://morphadorner.northwestern.edu/
[8] Verbix: http://www.verbix.com/

Finding the gender of a word: This is necessary in some languages (such as Romanian) and can be done by consulting a lexical database (RoWordNet for instance).

We evaluated our technique on 15 ontologies (from *physics*, *artificial intelligence*, *biology*, *mathematics*, etc.) and obtained a precision of approximately 90-93% when the question asked was from one of the question domains listed above. The system could be extended by adding other question domains and thus improving its overall quality. The main problems we encountered were the following: (1) not all public ontologies are complex enough to generate reliable AIML categories; (2) an efficient method to combine information about one concept from several sources has to be implemented.

3 Conclusions

This paper aims to present how a question answering component can be successfully added to a real robot. The question answering component we have designed allows our robot to convert an ontology in OWL format to a file in AIML format that works with an open-source implementation of AliceBot. This method gave good results, because the knowledge base is quickly constructed in a flexible manner and using existing information on the Internet. Currently the access to the question answering component is done through a console that allows human users to send questions to the robot. The answer is provided in both text and audio formats, using the synthesizer implemented in Windows and available in .NET Framework. The tests we conducted on a Lego NXT 2.0 robot showed that the time required to learn a new ontology is approximately 2-3 minutes (download the file, AIML processing, etc.), while answering a question takes roughly 3 seconds. In the future we want to implement a method to interpret voice commands, in order to address questions or to establish verbal communication between two robots.

References

[1] E. Freese. *Enhancing AIML Bots Using Semantic Web Technologies*. In Proc. Of Extreme Markup Languages (2007).

[2] R. Sumbaly, A. Kumar, G. Paruthi, S. Malhotra. *Artificially Intelligent Grid Assistant (A.I.G.A)*. Intern. Conf. on High Performance Computing (2007)

[3] R. Wallace. *Artificial Intelligence Markup Language (AIML)*. Working draft. A.L.I.C.E. AI Foundation (2005)

Tudor-Alexandru Dobrilă, Adrian Iftene