Original scientific paper

Marvin – A Conversational Agent Based Interface for the Study of Information Sciences

Nives Mikelić Preradović
Department of Information Sciences
Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10 000 Zagreb, Croatia
nmikelic@ffzg.hr

Damir Boras
Department of Information Sciences
Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10 000 Zagreb, Croatia
dboras@ffzg.hr

Sanja Kišiček
Department of Information Sciences
Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10 000 Zagreb, Croatia
smatic@ffzg.hr

Summary

In this paper we present initial results of the ongoing project, building a Conversational Agent (chatterbot) - Marvin. Marvin is designed to simulate intelligent conversation with students of Information Sciences at the Faculty of Humanities and Social Sciences in Zagreb. It is capable of providing basic feedback to students via textual methods.
The primary goal of this work is to inform the user of points of interest, to provide support, capture data from the user and promote study of information sciences. Furthermore, the goal is to enhance the presentation of information to students, especially information regarding the undergraduate study, obligatory and elective courses, ECTS (European Credit Transfer System) points and exams. Finally, the objective is to teach the students how to improve the quality of user experience using human-like conversational agents.

Key words: Conversational agents, chatterbots, AIML, Information Sciences curriculum, ECTS points

Introduction

Chatterbots are virtual characters capable of engaging a human counterpart in a meaningful conversation, often encountered as interfaces to help systems and

web-based search engines. They are systems that have the ability to parse natural language questions and, by referring to a knowledge base, generate natural language answers.

One of the most popular uses of the chatterbots is in the online tutoring process, by the application of online tutoring system based on dialogue [4], [5]. These chatterbots offer the possibility of control of the tutoring process itself, by monitoring the course of the learning process, recording the major characteristics of the process, and finally correcting it.

Although current systems lack the communication capabilities of the real human being, it has been shown [6] that even fairly simple chatterbots can increase the quality of experience for the user of interactive services and applications. In this work we explore the possibility of enhancing the quality of experience for the student using a virtual tutor Marvin.

The first version of Marvin is available at the following web address[1]: http://pandorabots.com/pandora/talk?botid=f98d56804e374cba.

Marvin was not designed to be a tutor that should actually teach particular subject matter, but to give an impression of a personal and human service that would increase the students' interest in the Study of Information Sciences, ease the search for particular information and boost satisfaction with the presentation of information.

We begin with a brief summary of related work and continue with an overview of AIML programming language, followed by the description of the system and its behavior. Finally, we discuss our future work and planned improvements to the system.

Related work

For the last decades, computer users have been witnessing new paradigms in human computer interfaces technology. The arrival of new gadgets and technologies is boosting the realization of friendly and easy-to-use interfaces.

Many systems for the English language have already been developed in this research area, but none for the Croatian language. A few of the chatterbots for English are presented in this chapter.

ELIZA [7] was the first chatterbot written by Joseph Weizenbaum between 1964 and 1966. It was a simulation of a Rogerian psychotherapist operated by processing users' responses to scripts. It rephrased the user's statements as questions and posed those to the "patient".

PARRY [2] was another famous early chatterbot who attempted to simulate a paranoid schizophrenic. It was designed in 1972 by psychiatrist Kenneth Colby.

[1] http://www.pandorabots.com is a software robot hosting service that allows to create and publish robots on the web from any browser. Together with Oddcast Inc.'s VHost™ platform, it allows publishing of flash-based interactive characters onto web sites, Intranets and mobile devices.

Although today one can find chatterbots that act as very advanced tutoring systems with sophisticated knowledge bases [1], most of the existing systems are focused on the quality and accuracy of information and the way it is presented to users, rather than the production of the knowledge base.
AIML is the XML dialect developed by Richard Wallace and a worldwide free software community between 1995 and 2002. It formed the basis for what was initially a highly extended Eliza called "A.L.I.C.E." (Artificial Linguistic Internet Computer Entity), which won the annual Loebner Prize Contest[2] for Most Human Computer three times and became the Chatterbox Challenge Champion in 2004.
The most popular online chatterbots based on AIML are Eliza[3] (both English and German version), Cypher[4](first Persian AIML bot), iGod[5], Kyle[6] (artificial intelligence bot which employs contextual learning algorithms), Shakespeare[7] bot, Ailis[8] (Italian bot) and Prelude[9] (a self learning bot with AIML support).

System description

This paper introduces the concept of using virtual human characters to provide support, capture data from the user and promote the Study of Information Sciences. To realize the concept, a virtual chatterbot, named Marvin, is developed for the students of Information Sciences at the Faculty of Humanities and Social Sciences. Marvin offers a simple and user friendly interface.
Marvin borrowed its name and personality from the paranoid android, a fictional character in *The Hitchhiker's Guide to the Galaxy* novel by Douglas Adams.
Since the basic idea was to build a chatterbot that will serve as an information source for the undergraduate students at the Department of Information Sciences, the biggest part of its knowledge base is information on obligatory and elective departmental courses from all six semesters of study.

[2] The scientific point of the Loebner Prize Competition is not to fool the judges (as it is usually stated in the literature), but to design a candidate that has indistinguishable performance indistinguishable to any judge [3].

[3] http://www.denkwerkzeuge.ch/

[4] http://www.syavash.com/portal/projects/cypher-yahoo-messenger-bot-project

[5] http://www.titane.ca/concordia/dfar251/igod/main.html

[6] http://www.leeds-city-guide.com/kyle

[7] http://www.shakespearebot.com/

[8] http://ai-tech.com/showcase/

[9] http://prelude.lennart-lopin.de/

AIML (Artificial Intelligence Markup Language) syntax

Firstly, the AIML template was built and served as a basic structure for all the courses and was used and adjusted by all students working on this project.

The basic unit of knowledge in AIML is called a category. Each category consists of an input question, an output answer, and an optional context. The question, or stimulus, is called the pattern. The optional context portion of the category consists of two variants, called `<that>` and `<topic>`. The tag `<that>` appears inside the category and its pattern has to match Marvin's last utterance. Remembering one last utterance is important if Marvin asks a question. This tag was implemented to use the user's reply to point the conversation in the specific direction.

The `<topic>` tag appears outside the category, and collects a group of categories together. The topic may be set inside any template.

The AIML pattern language consists of words, spaces and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. The pattern language is case invariant. Words are separated by a single space, and the wildcard characters function like words.

AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories.

All categories with information regarding the specific course are grouped in `<topics>` that are reached by input string "*kolegij*" followed by the acronym of the specific course, e.g. *kolegij OZ* for the course *Organizacija znanja.* If the user needs the access to the list of acronyms, he has to ask Marvin for the specific semester of study in order to get the acronyms for that semester.

If the user wants to abort the conversation about the specific course, he exits the topic with the command *nova tema.*

In order to anticipate as many user's answers as possible in the input patterns, the synonyms (such as nastavnik, profesor, predavač) in all 7 cases of singular and plural were often used as keywords, as well as wildcards * and _.

Recursion in AIML

Marvin produces the same reply to many different query formulations that share the same or similar meaning. This was achieved by recursion technique.

When building Marvin, we decided to reduce many ways of saying the same thing to one category, which contains the reply:

```
<category>
  <pattern>NASTAVNIK</pattern>
  <template>OIT predaje Hrvoje Stančić. Zanima te što još
on predaje? </template>
</category>
```

```
<category>
  <pattern>_ PREDAJE </pattern>
  <template> <srai>NASTAVNIK</srai> </template>
</category>
<category>
  <pattern>PREDAVAČ</pattern>
  <template> <srai>NASTAVNIK</srai> </template>
</category>
<category>
  <pattern>_ DRŽI </pattern>
  <template> <srai>NASTAVNIK</srai> </template>
</category>
<category>
  <pattern>PROFESOR</pattern>
  <template> <srai>NASTAVNIK</srai> </template>
</category>
```

AIML implements recursion with the <srai> operator, so that the output depends not only on one matched category, but also any other recursively reached through <srai>.
There is a variety of applications for <srai> in AIML, since it is used to reduce complex grammar forms to simpler ones, to split the input into two or more subparts and combine the responses to each or to map different ways of saying the same thing to the same reply (synonyms). Also, this operator is used for spelling or grammar corrections and for detecting keywords anywhere in the input.

Marvin's knowledge base and reply structure

The course knowledge base covers the name of the lecturer, the course prerequisites (if any), the number of ECTS points for the specific course, the hours of lectures / seminars / labs per course, the status of the course (obligatory or elective) and the description of the exam. Also, Marvin gives information if a specific course is prerequisite to some other course or prerequisite to the graduate study. It also has built-in information regarding the profile of undergraduate study (such as the required number of ECTS points for each semester of the study, etc).
Apart from that, Marvin has built-in information regarding the Department profile, history and staff (names of the head of the Department, deputy head of the Department, Department administrator, Department librarian, working hours of the Department administration and library).
While coding this information, we did not use <topics> in AIML, but only <categories> that contain the corresponding keywords from the most probable user's queries in their entry patterns.

When providing the reply about the name of the teacher/lecturer, Marvin tries to boost communication with the user posing the question to the user such as: *are you interested in other courses taught by this lecturer?*
The communication flow from this point onwards depends on the user's answer (this part of communication is coded with the tag `<that>`).
Since the same question can be formulated in many different ways and since we cannot predict that user will pose the question in most frequent or in the shortest way, we had to use the wildcards * and _ with extreme caution. Therefore, for some keywords we decided to introduce several combinations with * and _, depending on the words preceding and/or following the keyword.
The exact content and the structure of Marvin's reply are different in two levels:
1) Every topic change triggers the content of the reply in a way that it attempts to follow the normal flow and nature of the communication (e.g. if user changes the name of the course in his utterance, Marvin will immediately change the content of the reply, providing the user only with the necessary information)
2) Most of Marvin's replies are enhanced with our personal comments, which are usually slightly humorous. The comments are designed with a goal to convince the user that the chatterbot is a student himself, who took all or at least some of the courses. The example of the comment is: *"Oh, in my time it was different…", "When I was taking this exam, it was 12 pages long, can you believe it?"*

Implementing keywords
Our goal was to write an AIML template which can be activated by the appearance of a keyword anywhere in the input sentence. The AIML categories that are built into Marvin are illustrated by the following example:

```
<category>
<pattern>TKO _ TAJNICA</pattern>
<template>Tajnica odsjeka je Nevenka Petak.</template>
<category>
<pattern>_ TKO _ TAJNICA </pattern>
<template><srai> TKO _ TAJNICA </srai></template>
</category>
<category>
<pattern> TKO _ TAJNICA _</pattern>
<template><srai> TKO _ TAJNICA </srai></template>
</category>
<category>
<pattern>_ TKO _ TAJNICA *</pattern>
<template><srai> TKO _ TAJNICA </srai></template>
</category>
```

The first category both detects the keyword when it appears by itself and provides the generic response. The second category detects the keyword as the suffix of a sentence. The third detects it as the prefix of an input sentence, and finally the last category detects the keyword as an infix. Each of the last three categories uses `<srai>` to link to the first, so that all four cases produce the same reply, but it needs to be written and stored only once.

Conversation context

In AIML syntax `<that>...</that>` encloses a pattern that refers to the bot's previous utterance. Specifically, if the chatterbot responds with a multiple sentence paragraph, the value of `<that>` is set to the last sentence in the sequence. A common application of `<that>` is found in yes-no questions:

```
<category>
  <pattern>DA</pattern>
    <that>ZANIMA TE ŠTO JOŠ PROF. TUDMAN PREDAJE</that>
  <template>Na preddiplomskom studiju prof. dr. sc. Tudman
predaje osim OZ-a i TIZ.
  </template>
</category>
```

This category is activated when the client says YES. The chatterbot must find out what he is saying "*yes*" to. If the bot asked, *"Are you interested in what prof. Stancic is teaching apart from this course?*" this category matches, and the response, *"Apart from OIT, he teaches the part of the ODOTIS course, as well as Informatics for Archaeologists",* continues the conversation along the same lines.

The AIML interpreter stores the input pattern, `<that>` pattern and `<topic>` pattern along a single path (e.g. `INPUT <that> THAT <topic> TOPIC`). When the values of `<that>` or `<topic>` are not specified, it implicitly sets the values of the corresponding of `<that>` or `<topic>` pattern to the wildcard *.

The first part of the path to match is the input. If more than one category have the same input pattern, the system will distinguish between them depending on the value of `<that>`. If two or more categories have the same `<pattern>` and `<that>`, system will choose the reply based on the `<topic>`.

Symbolic reduction

Symbolic reduction refers to the process of simplifying complex grammatical forms into simpler ones. Usually, the atomic patterns in categories storing Marvin's knowledge are stated in the simplest possible terms. In other words, we tend to prefer patterns like "*WHO IS THE HEAD OF THE DEPARTMENT*" to ones like *"DO YOU KNOW WHO THE HEAD OF THE DEPARTMENT IS*" when storing information about the head of the Department.

Many of complex forms were reduced to simpler forms using symbolic reduction:

```
<category>
<pattern>DO YOU KNOW WHO * IS</pattern>
<template><srai>WHO IS <star/></srai></template>
</category>
```

Whatever input matched this pattern, the portion bound to the wildcard * may be inserted into the reply with the markup `<star/>`. This category reduced any input of the form "*Do you know who X is?*" to "*Who is X?*"

One of the most useful applications of `<topic>` are subject-dependent "pickup lines", like:

```
<topic name="mediji">
<category>
<pattern>*</pattern>
<template>
<random>
<li>Gledas li televiziju?</li>
<li>Koje novine citas?</li>
<li>Koji radio slusas?</li>
<li>Koje portale pratis?</li>
</random>
</template>
</category>
```

Finally, apart from the above mentioned functions, Marvin also acts as a language tutor and corrector of spelling and grammar mistakes that students make. Here is the example of that function:

```
<category>
<pattern>UVIJET *</pattern>
<template>Mislim da si htio reći uvjet, zar ne?
</template>
</category>
```

Evaluation

Marvin's evaluators were four most successful undergraduate trainee teachers in the final year of the undergraduate study at the Department of Information Sciences who tested out the knowledge base through dialogue with the chatterbot. They checked the database for accuracy, interpretation and relevance to the types and the level of questions being asked.

The discussion consisted of 50 inputs and answers and each evaluator spent 30 minutes on average chatting with the chatterbot. A different set of input sentences was used by each evaluator.

The evaluators chatted with Marvin from two perspectives: as the competent information scientists that they were, as well as pretend bachelor students in order to see how well the chatterbot interacted in the specific scientific field and overall knowledge.

Analyses were conducted post-hoc using the transcripts saved by the evaluators from all of their chat sessions.

From the pretend bachelor students' perspective, trainee teachers concluded that the chatterbot works best with single clause utterances, each exchange being treated virtually independently. When they moved to multi clause units, or look at exchanges which range over more than one turn, Marvin's limitations become much more apparent. In other words, although Marvin's database contains some very basic factual world knowledge, the depth of knowledge is too narrow to cope with open-ended conversations with humans.

From the competent information scientists' perspective, trainee teachers evaluated Marvin's answers based on the following categories: good answers, reasonable answers, and off topic answers, that seem to have little or nothing to do with the input clause.

Although 47% of the answers were classified as good answers and although 41% of the answers were classified as reasonable answers, evaluators concluded that clusters of knowledge about different topics that have been input into Marvin's knowledge database are at a somewhat superficial level, since Marvin has no actual knowledge of what it is talking about and cannot discuss a topic.

Since Marvin draws from his general knowledge database in terms of heuristics to avoid answering a question it in fact has no answer for, and suggests a new topic, giving the illusion that it actually has something to say about the new topic, the evaluators suggested improving Marvin's database with more factual world knowledge.

Conclusion and future work

As part of our future work, we plan to modify Marvin to take on the personality and knowledge base of named individuals. If a sufficiently high-quality knowledge base is constructed, then there is scope in higher education for using such a chatterbot as a substitute for the expert academic.

Many students of information sciences experience problems in learning theory for the specific courses in both undergraduate and graduate studies. They usually find it very hard to get started with reading, need cartoon level introduction, leading on to more complex material and they find it hard to know what is relevant discussion, so seminar discussion often wanders off topic.

In this context, an accessible chatterbot with a knowledge base reflecting key areas of information science could make an important educational contribution.

In our future work we plan to develop a small number of knowledge bases for use in information science study, particularly in Knowledge Organization course and Theory of Information Science course. These will consist primarily of personality and knowledge linked with key information theorists. The knowledge bases will be designed to be sensitive to the knowledge levels of potential student users and to be open to explanatory questions. Such a chatterbot can be used as a general advisor or expert. Students who have a research question or

essay title that needs to be researched could use the bot to generate content that can be included in their assessed work. Conversation with the bot will be recordable and can then be cut and pasted into an essay. Students will be required to edit this information into a coherent essay, just as they would with information collected from texts.
Such chatterbot would use Wikipedia information to build its conversations and would offer links to Wikipedia articles in the field of information science that students wrote and submitted to Wikipedia. The result would be the development of AIML chatterbot that would use key information science knowledge bases and provide extra services and linked information website. This should produce a better understanding both of the form and type of content that best matches student needs and also information about the best ways in which the bots can be used educationally.

Acknowledgement

This work was partly carried out within the research projects “Croatian Dictionary Heritage and Croatian European Identity” and “Public Knowledge Design and Management in Information Space” supported by the Ministry of Science, Education and Sports of the Republic of Croatia. Much of the work was done as a student project at the Faculty of Humanities and Social Sciences in Zagreb, Croatia.
The implementation and modeling was done by undergraduate students: Jasna Turković, Mirjana Horvacki, Eva Cukor, Bojan Kopitar, Tomo Šala and Julija Maksimović.

References

Burleson, W. Affective Learning Companions: strategies for empathetic agents with real-time multimodal affective sensing to foster metacognitive and meta-affective approaches to learning, motivation, and perseverance : PhD thesis. Massachusetts Institute of Technologies, 2006

Guzeldere, G.; Franchi, S. Dialogues with colorful personalities of early AI. // *Constructions of the Mind: Artificial Intelligence and the Humanities*. volume 4 (1995), issue 2

Harnad, S. The Turing Test Is Not A Trick: Turing Indistinguishability Is A Scientific Criterion. // *SIGART Bulletin*. volume 3 (October, 1992), number 4; pp. 9-10

Kovacic, B.; Skocir, Z. Development of Distance Learning System Based on Dialogue. // *Proceedings of the IEEE Region 8 Conference EUROCON 2003 : Computer as a tool.* / Zajc, B.; Tklacic, M. Ljubljana, Slovenia : Faculty of Electrical Engineering, University of Ljubljana. volume 1 (22-24 Sept. 2003); pp. 224-228

Kovacic, B.; Skocir, Z. Formal Model for Distance Learning System Based on Dialogue. // *Proceedings of the International Conference ICT2001*. Bukurešt. volume 1 (July 2001); pp. 231-236

Massaro, D. W., In Spencer, P.E.; Marshark, M. (Eds.). A computer-animated tutor for language learning: Research and applications. Advances in the spoken language development of deaf and hard-of-hearing children (pp. 212-243). New York, NY: Oxford University Press, 2006

Weizenbaum, J., ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine. // *Communications of the ACM*. volume 9 (1965), issue 1; pp. 36-45