First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A multilingual conversational agent with speech recognition and text-to-speech capabilities

Mihai Stancu

Abstract

Conversational agents are computer programs that utilize natural language technologies to engage users in human-like, text-based, information-seeking and task-oriented "dialogs". They can support a broad range of applications in business enterprises, education, government, healthcare, and entertainment. In this paper we propose and implement a conversational agent based on AIML, with speech recognition and text-to-speech capabilities and a virtual avatar. We describe the challenges we had to face in order to improve the artificial intelligence, and the quality of the speech synthesis for the Romanian language.

1 Introduction

The most natural form of communication is face-to-face communication. This is why using virtual avatars and natural speech is very useful for human-computer interaction [1]. Many artificial conversational entities or conversational agents, speech recognition and text-to-speech engines, as well as virtual 2d or 3d characters have been created for this purpose [1],[2],[3],[4]. Today there are numerous applications of conversational agents in the following areas [3],[5]:

- businesses:
  - customer service: responding to customers' general questions about products and services
  - help desk: responding to internal employee questions
  - website navigation: guiding customers to relevant portions of complex websites.
  - guided selling: providing answers and guidance in the sales process
  - technical support: responding to technical problems
- telecommunications: eliminating the need for human-assisted support
- education: creating virtual tutors and trainers for interactive learning
- healthcare: creating information experts and virtual communication partners for therapy
- entertainment: creating life-like videogame characters and virtual actors

The main goal of this paper is to propose and implement a multilingual conversational agent with speech recognition and text-to-speech capabilities. Though there are many similar applications, the majority of which are clones after the original, what this paper brings to this field is a new way of utilizing existing technologies to achieve a better human-computer interaction.

The paper has the following contents: section 2 presents a short classification of the different conversational agents up to date, section 3 describes shortly the systems used in order to reach our goal, in section 4 we have the design and implementation of the conversational agent, and finally, in the last section we give our results and a plan for future developments.

2 The main types of conversational agents

Conversational agents can be classified by the following categories [3],[4]:

1. The technology used for the conversational engine (AIML, bayessian networks, etc.)
2. The degree of interactivity (text-based interface, SR and TTS capabilities, avatars, etc.)
3. The goal for which it was created (the applications mentioned in the first section)

If the first agents only allowed a text-based console for the conversation to take place, once the technology developed, new ways of interaction based on speech services and 2D and 3D graphics have evolved, reaching closer and closer to true human-like appearances.

More advanced means of comparing embodied conversational agents consist in evaluating the users' reactions and emotional responses while interacting with the agent [4]. In other words, it is a way of measuring the realism of the conversational agent, similar to the famous Turing test [6].

In 1990 Hugh Loebner agreed with The Cambridge Center for Behavioral Studies to underwrite a contest designed to implement the Turing Test. Dr. Loebner pledged a Grand Prize of $100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's. Each year, a prize of $2000 and a bronze medal is awarded to the most human-like computer. The winner of the annual contest is the best entry relative to other entries that year. [7]

3 Technologies used for developing the conversational agent

We have used three technologies that have proved their value through the extremely large number of applications that utilize them today. We will now shortly present these technologies and in the following section we will show exactly how we combined them and came up with new techniques in order to achieve our goal.

Additionally we have used software applications such as the trial versions of Adobe Photoshop and Adobe Flash Professional in order to create the graphics files, the gestures and lip-sync animations of our avatar.

3.1 AIML

AIML (Artificial Intelligence Markup Language) is an XML compatible, easy to learn language, that facilitates the rapid creation of a knowledge base for a conversational agent. [8] Developed by Richard Wallace and a worldwide free software community between the years of 1995 and 2002, it formed the basis for A.L.I.C.E ("Artificial Linguistic Internet Computer Entity"), which won the annual Loebner Prize Contest for Most Human Computer three times, and was also the Chatterbox Challenge Champion in 2004 [9].

Because the A.L.I.C.E. AIML set was released under the GNU GPL, and because most AIML interpreters are offered under a free or open source license, many "Alicebot clones" have been created based upon the original implementation of the program and its AIML knowledge base, and this fact has eased very much the development of our own application [10]. Free AIML sets in several languages have been developed and made available by the user community. It was these sets that made possible for our application to support multiple languages.

The most important AIML tags are:

<aiml> which marks the beginning and the end of an AIML document;

<category> which marks a unit of the knowledge base;

<template> which marks the response to a question.

There are over 20 other tags which can be used, that allow the formation of so-called personalized predicates with the help of wildcards. Also, it is possible for a category to be called by another, and the language allows for the directing of a conversation to a specific topic.

Other advantages consist in using recurrence with the help of the <srai> operator. Without going into too much detail, we list the other possibilities the language has to offer: symbolic reduction, divide et impera, synonym substitution, keyword detection and implementation of conditionals.

3.2 MBROLA

MBROLA (Multi Band Re-synthesis Overlap-Add) is an algorithm for speech synthesis, and software which is distributed at no financial cost but in binary form only.

Initiated by the TCTS Lab of the Faculté Polytechnique de Mons in 1996, MBROLA is now a worldwide collaborative project. The MBROLA project web page [11] provides resources for a large number of spoken languages and voices.

The MBROLA software is not a complete text-to-speech system for all those languages. The text must first be transformed into phoneme and prosodic information in MBROLA's format. The input consists of a list of phonemes, together with the duration of the phonemes and a piecewise linear description of pitch, and produces speech samples on 16 bits, at the sampling frequency of the diphone database used. Separate software to do this is available for some but not all of MBROLA's languages This is the reason we implemented our own TTS system that supports multiple languages and developed for the first time a pipeline for processing raw text for Romanian speech synthesis, which will be discussed in the following section of our paper.

Although diphone-based, the quality of MBROLA's synthesis is considered to be higher than that of most diphone synthesizers. This is due in part to the fact that it is based on a preprocessing of diphones (imposing constant pitch and harmonic phases), which enhances their concatenation while only slightly degrading their segmental quality [12].

MBROLA is a time-domain algorithm, as PSOLA (Pitch Synchronous Overlap-Add) created by Paul Taylor [13], which implies very low computational load at synthesis time. Unlike PSOLA, however, MBROLA does not require a preliminary marking of pitch periods. This feature has made it possible to develop the MBROLA project around the MBROLA algorithm, through which many speech research labs, companies, or individuals around the world have provided diphone databases for 34 languages, a number which is by far a world record for speech synthesis.

3.3 MS Agent

Microsoft Agent is a set of software services that enable developers to incorporate interactive animated characters into their applications. [14] These characters can speak, via a text-to-speech engine or recorded audio, and accept spoken voice commands through speech recognition.

Microsoft Agent was first introduced through Microsoft Bob in 1995, which used an early version of Agent technology internally referred to as "Microsoft Actor". Microsoft Agent became popular as the initial version of the Office Assistant in Office 97, sometimes dubbed "Clippit" or "Clippy". [15] The first version of Microsoft Agent was released on MSDN in 1998 and the MS Agent version 2.0 core components were available for download on Microsoft’s website in 2003.

Microsoft Agent includes an ActiveX control that makes its services accessible to any programming language that supports this type of control. This means that interaction with the characters can be programmed even in HTML.

The speech engine used by MSAgent is driven by the Microsoft Speech API (SAPI), version 4 and above. Microsoft SAPI provides a control panel for easily installing and switching between various available text-to-speech and speech recognition engines, as well as voice training and scoring systems to improve quality and accuracy. Because the MBROLA voices are not compatible with SAPI, one of our future goals is to create a SAPI compliant TTS engine for all 34 languages available on MBROLA’s website.

Microsoft Agent characters are stored in files with the .ACS extension, and can be stored in a number of compressed .ACF files for better Internet distribution. Microsoft provides four agent characters for free, which can be downloaded from the Microsoft Agent website. These are called Peedy, Merlin, Genie, and Robby. New Agent characters can also be created using Microsoft's development tools, including the Agent Character Editor which we used for creating the character for our application.

4 Designing and implementing the conversational agent

In this section we present the main algorithms we used and the difficulties we had to surpass in order to make our program as modular and as flexible as possible.

The system is implemented in C# and works as follows: the user inputs text from the keyboard which is then compared to one of the predefined commands. Then the text is sent to the AI module. This module uses the AIML language to store a database of knowledge, in order to generate a response. The response is then sent to the TTS module which will perform the speech synthesis. Once the generated audio file is ready to be played, a corresponding animation sequence for the virtual avatar is rendered. The process repeats itself until termination of the program. These steps are illustrated in figure 1.

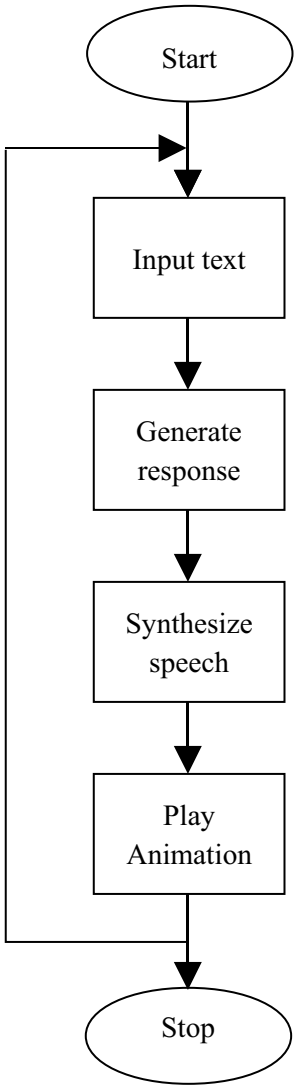

Figure 1. Functioning principle of the conversational agent

Because there is no freely available SAPI compliant Romanian TTS engine, the most difficult part was to create such an engine. In the following lines we describe how our TTS system works. Inspired by [16],[17],[18], the engine contains a natural language processing mechanism. Text is transmitted to a pre-processor, then the program generates phoneme and prosodic information and finally, the MBROLA system is called with these parameters and generates the speech audio files. These steps are illustrated in figure 2.

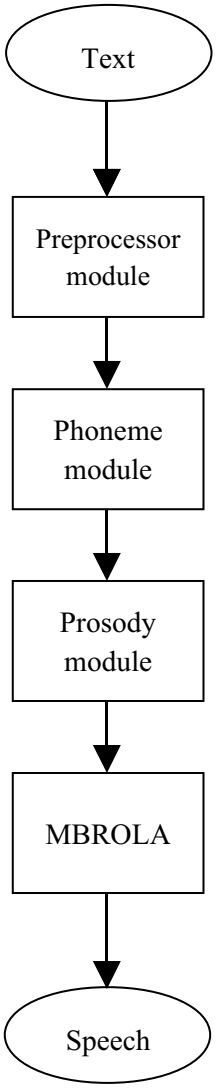

Figure 2. Functioning principle of the Romanian TTS engine

The architecture of this TTS engine allows multiple language support. Because the application we are building is a work in progress, there is no compatibility with the Speech Application Interface from Microsoft yet. In order to take advantage of SAPI, we can use other Microsoft voices and languages for speech synthesis until the final release of our conversational agent. In the next lines we give further details about every module of our TTS engine, along with a few examples written for the Romanian language, to better describe what is happening at each step.

4.1 Preprocessor module

This module uses an abbreviations dictionary and also a symbols dictionary in order to substitute all abbreviations, mathematical symbols, punctuation marks, etc. with the corresponding words or phrases in the current language. This module also replaces all Arabic and Roman numerals, all the times and dates with the corresponding text strings and changes all uppercase letters to lowercase.

As an example, the statement: "În data de 27.08.2003 locuiam pe str. Pieții, bl. 18, sc. C, et. IV, or. Victoria, jud. Brașov, România" will become "în data de douăzeci și șapte august două mii trei locuiam pe strada pieții bloc optisprezece scara ce etajul patru orașul Victoria județul brașov românia".

4.2 Phoneme module

The phoneme module is responsible for coding the preprocessed text in a sequence of phonemes. The phoneme symbols are the same SAMPA (Speech Assessment Methods Phonetic Alphabet) symbols used for the diphone database of a certain language. In order to do this, we use a dictionary of phonetic rules and a dictionary of phonetic exceptions. For the Romanian language these dictionaries have quite a small file size, unlike for English or other languages, because the Romanian language is mostly a phonetic language, i.e. words are spelled as they are written. This is also the reason why Romanian speech synthesis has a better quality than for other languages.

4.3 Prosody module

This represents the final step of our natural language processing pipeline. The prosody module gives the correct intonation for every word of the phonemized text output from the previous module. In the case of the Romanian language, we can apply a few simple rules in order to give the voice more naturalness. First every word is separated into syllables and the last but one is accented, i.e. the pitch for the sounds corresponding to that syllable is increased. Of course, not all the words in the Romanian language are accented this way, and thus a dictionary of accenting exceptions helps to accent all the words in the correct manner. This module also determines the speed at which the text is "read". By modifying the duration of the sounds corresponding to every phoneme, we can adjust the rate of speech, and also increase or shorten the pause between words. Another aspect that is currently in development is creating emotional states for the voices, again by changing the pitch values of the generated sounds, according to specific, predetermined patterns.

In figure 3 we show, for clarification, the contents of the file bonjour.pho supplied as an example for the French FR1 diphone database.

```
_ 51 25 114
b 62
o~ 127 48 170
Z 110 53 116
u 211
R 150 50 91
_ 91
```

Figure 3. Contents of bonjour.pho file

This shows the format of the input data required by MBROLA. Each line contains a phoneme name, a duration (in ms), and a series (possibly none) of pitch pattern points composed of two integer numbers each : the position of the pitch pattern point within the phoneme (in % of its total duration), and the pitch value (in Hz) at this position.

Hence, the first line of bonjour.pho : _ 51 25 114 tells the synthesizer to produce a silence of 51 ms, and to put a pitch pattern point of 114 Hz at 25% of 51 ms.

5 Conclusions and future developments

In this paper we have presented the design and implementation of a multilingual conversational agent with speech recognition and text-to-speech capabilities. Numerous efforts have been made in order to increase the interactivity and the quality of the speech synthesis for the Romanian language. Nonetheless there are many things that can be improved, that are planned for future versions of our application. Among these we mention the following:

- extending the supported languages and the knowledge base;
- incorporating a logic module, an arithmetic module, and a learning module;
- creating an agent editor and new virtual characters

References

[1] A. Paiva (ed.). *Affective Interactions: Towards a New Generation of Computer Interfaces*. Springer, Berlin, 2000.
[2] T. Rist, R. Aylett, D. Ballin, J. Rickel (eds.). *Intelligent Virtual Agents*. Springer, Berlin, 2003.
[3] H. Predinger, M. Ishizuka (eds.). *Life-Like Characters: Tools, Affective Functions, and Applications*. Springer, Berlin, 2004.
[4] J. Cassel, J. Sullivan, S. Prevost, E. Churchill. Embodied Conversational Agents. MIT Press, Cambridge, Massachussetts, 2000.
[5] J. Lester. *PracticalHandbook of Internet Computing*. CRC Press, Boca Raton, 2009.
[6] http://en.wikipedia.org/wiki/Turing_test
[7] http://www.loebner.net/Prizef/loebner-prize.html
[8] http://www.alicebot.org/aiml.html
[9] http://en.wikipedia.org/wiki/AIML
[10] http://aimlbot.sourceforge.net
[11] http://tcts.fpms.ac.be/synthesis/mbrola.html
[12] http://en.wikipedia.org/wiki/MBROLA
[13] P. Taylor. *Text-to-Speech*. Cambridge University Press, Cambridge, 2009.
[14] http://www.microsoft.com/products/msagent/main.aspx
[15] http://en.wikipedia.org/wiki/Microsoft_Agent
[16] D. Burileanu, C. Dan, M. Sima, Mihai, *A Parser-based Text Preprocessor for Romanian Language TTS Synthesis,* Eurospeech'99, Budapest, 1999
[17] O. Buza, G. Toderean, *A Romanian Syllable-Based Text-to-Speech System,* 6th WSEAS International Conference, Corfu Island, 2007
[18] A. Ferencz, T. Ratiu, M. Ferencz, T.C. Kovacs, I. Nagy, D. Zaiu, *Romvox – Experiments Regarding Unrestricted Text-to-Speech Synthesis for the Romanian Language,* 6th EWONLG, Duisburg, 1998

Stancu Mihai
Universitatea Lucian Blaga din Sibiu
Facultatea de Stiinte
Bd-ul. Victoriei, Nr.10, Sibiu
Romania
stancu_meehigh@yahoo.com