

# The Artificial Librarian

## A Database Dialogue System

Mattias Johansson and Jonas Håkansson  
*Department of Computer Science*  
*Lund Institute of Technology*  
*Lund University, Sweden*

January 17, 2003

### 1. Introduction

This project is the second part of the course “An Introduction to Language Processing and Computational Linguistics” at the Department of Computer Science. It was defined by us together with our teacher Pierre Nugues. The Artificial Librarian is a dialogue system which is connected to a database. It parses the user input and searches the database for the information. The user can for example ask for a book by a specific author, who has written a specific book or where in the library the book is located. The librarian can sometimes ask for additional information, for example to narrow down the search. The application is web based and can be found at <http://www.efd.lth.se/~e98mj/librarian/>. It may not always be up and running though. There is a screenshot in appendix B to give an idea on how the system looks.

### 2. Technology overview

The application was decided to be connected to a public book database to make the query constraints as few as possible. Since the application was to be connected to the database through internet it felt natural to make an internet interface for the whole application. Our teacher suggested JSP, Java Server Pages. We had never used this technique, but most of our programming courses are based on Java, so it seemed like a good idea.

### 3. Language processing

The language processing consists of two parts; one that interprets the user input and one that generates the response from the librarian. The main technique we use for the user input interpretation is decision trees, which simplified works like this: classify the first word, and then take the next word and check if it fits with some of the parts-of-speech that are allowed after the first one. This way, word by word in the sentence is traversed. The dialogue processor, that generates the response from the librarian, is explained in chapter 3.3.

### 3.1 Parts-of-speech

When a word is to be classified it is compared to the list of words in the part-of-speech table. Examples of the words are shown in the table below.

| Part-of-speech | Examples                           |
|----------------|------------------------------------|
| Auxiliary      | do, can, has                       |
| Interjection   | yes, no, ok, okay                  |
| Noun           | shelf, books                       |
| Preposition    | by, on                             |
| Pronoun        | you, I, anything, something, any   |
| Proper noun    | any proper noun, e.g. Stephen King |
| Verb           | have, find, written, wrote         |
| Wh-word        | what, where, who, which            |

### 3.2 Possible decision trees

The tree consists of part-of-speech phrases. These phrases consist of words of the same part-of-speech. Which phrases or words that are allowed as the next, are shown in the table below. Every sentence begins with the sentence phrase. The first word of a sentence is compared to the list in the second column in the table. If it fits, the next word is checked in the same way. Not all parts-of-speech are allowed as a last word in a sentence. The rules are shown in the third column. Since the phrases not are words, there is no information on those rows. As given from that column the only two parts-of-speech that are allowed as last word are proper nouns and interjections. There is a special case when the last word not is a proper noun or an interjection. When the librarian has asked for something and the user answers with “yes, [phrase]”, the phrase part is interpreted, but if the phrase part doesn’t fit any of the possible types of sentences, the phrase part is neglected and the sentence is interpreted as only “yes”. This means that all words are allowed as a last word if it is a phrase that follows a “yes” or “no” and the phrase couldn’t be interpreted.

| Root or current leaf | Possible subsequent leafs  | Allowed as last word |
|----------------------|--|----------------------|
| Sentence (phrase)    | wh-word, preposition phrase, interjection, proper noun phrase, auxiliary | -                    |
| Noun phrase          | pronoun, noun  | -                    |
| Preposition phrase   | preposition  | -                    |
| Proper noun phrase   | proper noun  | -                    |
| Verb phrase          | verb   | -                    |
| Auxiliary            | noun phrase, verb phrase   | no                   |
| Interjection         | proper noun phrase, preposition phrase, wh-word                          | yes                  |
| Noun                 | auxiliary, preposition phrase  | no                   |
| Preposition          | proper noun phrase, wh-word  | no                   |
| Pronoun              | verb phrase, preposition phrase, noun phrase                             | no                   |
| Proper noun          | proper noun phrase, preposition phrase                                   | yes                  |
| Verb                 | preposition phrase, proper noun phrase, noun phrase                      | no                   |
| Wh-word              | auxiliary, noun phrase, verb phrase                                      | no                   |

### 3.3 Dialogue processor

The user input is searched for special words so that the system knows what the user wants. For example if the user begins the sentence with “who”, the system knows that the user is looking for authors. When a successful interpretation of the user input is done, the query is sent to the database. Otherwise, if it could not be interpreted, the librarian has to ask the user to type something else. When the answer is received from the database, and the result is extracted, the system can decide what the answer should be. The system now checks how many results the database returned and builds different answers based on that information. On some occasions the systems gives more information then the user asked for, e.g. on what shelf the books are located. All possible dialogues are listed in a tree structure in appendix A.

## 4. Information extraction from web database

The web database we decided to use was Amazon.com. This is an online book store which provides a huge amount of books, so that it will look like there are that many books in the library. The draw back of using a book store is that there are sometimes different editions of a book. If you for example ask “What do you have by Tolkien?”, the answer would be:

*I found more than 25 titles. The five most popular, I think, are:*

- ✘ *The Hobbit and The Lord of the Rings [BOX SET]*
- ✘ *The Lord of the Rings [BOX SET]*
- ✘ *The Lord of the Rings (Leatherette Collector's Edition)*
- ✘ *The Hobbit (Leatherette Collector's Edition)*
- ✘ *The Silmarillion*

*Is it any special book you are interested in?*

This is not the answer you expect when it is a library. It should only answer with the title and not with multiple versions of the same book.

### 4.1 Retrieve result from web page

Amazon.com only provides an html based version of the database available to people outside the company. This means that when we send our database queries we get an html page as an answer. Since Amazon.com is one of the largest online book stores, it has a lot of commercial banners and special offers on the site. This made it a little more difficult to extract the significant results from the answer. If Amazon.com doesn't find any book or it only finds one, the answer page looks completely different, but we managed to find some patterns to recognise the significant results, even in such cases. For example, there was a number on the page that told how many results that was found, but the number only showed if there were two or more results. Another potential problem with getting the result page in html format is that it shows a maximum of only 25 hits per page. As the system is built now that is not a problem, but if we sometime are going to develop it further it might be a limitation. To get the most likely answer when the user asks for an author or a title, we chosen to use Amazon.com's feature of sorting the results by best selling. Then, when the system suggests an author or a title it picks the top item of the list.

## 5. Development

To give an overview of the way our work proceeded, we have put together some milestones.

- **Selected techniques:** Since we have programmed a lot using Java earlier, we chose to use a Java based environment; Apache Tomcat which is a web server supporting JSP (Java Server Pages). As we run the server on our school account, we can not have it up and running when we are not logged on. We looked around for an online book store to use as the database for our project, and found that Amazon.com would be the best choice, even though it sometimes could be a little slow, because of its large number of books and the consistent way it presents the search results.
- **Extracted search results:** In the beginning we had some difficulties to decode the result pages, because of the large amount of commercials and special offers on the result pages, but managed to solve the problems and wrote code for searching by author, title and both.
- **The Language Processing module:** Initially we wrote code that could handle a couple of types of sentences. Before we decided to use decision trees to represent the allowed sentence structures, we tried a couple of other homemade variants. This was probably the part of the project that chewed the most time.
- **Connected the modules:** We developed the Language Processing module and the Web Database module separately to have better control over eventual misbehaviours. As we got them both to work independently, we connected them and for the first time got a working web application.
- **The dialog history:** To be able to get an overview of the dialog (since each turn can take quite a while to process, see selected techniques above), we added the history to returned web pages.
- **The multi-turn dialogs:** To make a more interesting application, we implemented multi-turn dialogs, so that the user would be able to give answers to the librarian's following-up questions. The recognized answers would be "yes", "no" and "okay". The user may also continue a "yes"-answer with a supported sentence that would be used if correct.
- **Conversation enders:** To reset the dialog and its history, we added conversation enders like "goodbye" and "thank you"
- **Map of library:** We added a simple map as a complement to the referred sections of the library. It makes the visualisation of the artificial library easier.
- **Testing the application:** We let people not involved in the project try The Artificial Librarian to get opinions about it, to see what kind of sentences they asked, and to find eventual bugs.

## 6. Examples

To give you a picture of how The Artificial Librarian works, we have provided a demonstration with four examples.

### Example 1:

**U:** Who has written Pippi Långstrump?

**L:** I found over 25 matching titles. Could it perhaps be Astrid Lindgren, Michael Chesworth (Illustrator)? Otherwise, you have to be more specific.

**U:** Yes, that's right

**L:** Great, I'm glad I could help you!

### Example 2:

**U:** What books do you have by King?

**L:** I found more than 25 titles. The five most popular, I think, are:

- ☒ From a Buick 8: A Novel
- ☒ Everything's Eventual: 14 Dark Tales
- ☒ On Writing: A Memoir of the Craft
- ☒ Brunelleschi's Dome: How a Renaissance Genius Reinvented Architecture
- ☒ The Foster's Market Cookbook

Is it any special book you are interested in?

**U:** Yes

**L:** Which one would that be?

**U:** Everything's Eventual

**L:** It would be to your left in the 'K' section.

### Example 3:

**U:** What do you have by Jan Guillou?

**L:** I found 18 titles. The most popular ones are:

- ☒ Enemy's Enemy
- ☒ Væagen till Jerusalem
- ☒ Tempelriddaren
- ☒ Irak--det nya Arabien
- ☒ Artister : intervjuer och porträtt

Is it any special book you are interested in?

**U:** Yes, on which shelf can I find Tempelriddaren?

**L:** It would be to your left in the 'G' section.

### Example 4:

**U:** Where can I find books by Långstrump?

**L:** I'm sorry, but it seems like we don't have anything by Långstrump.

**U:** Okay

**L:** What else can I do for you?

**U:** Nothing

## 7. Conclusions

The problem discussed in chapter 4, that the system answers with different versions of the same book, should be dealt with. If we some day would decide to develop The Artificial Librarian further, we would probably use a library database to make it more realistic. Another solution is that we could change the library to an information desk in a book store. This would be much easier of course, but it could have other features as price information and price limitation in a search.

When we started with this project we did not really know what result we could expect because we were all new to computational linguistics. We started with a not very high ambition level, but our teacher encouraged us to make it more and more extensive. Now when we are finished, the most important lesson we have learned is that it doesn't take a huge program to make an interactive dialogue system which work fairly well, although there are some linguistic tricks that make our system look smarter then it is.

Overall we are satisfied with the result of this project. Of course there are things that could be better, but it was only a two university points (two weeks) project.

## 8. References

J Boye, M Wirén, M Rayner, I Lewin, D Carter and R Becket, July 1999, **Language-Processing Strategies and Mixed-Initiative Dialogues**, IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems

S Abney, M Collins and A Singhal, 2000, **Answer Extraction**, AT&T Shannon Laboratory

P Nugues, 2002, **Lecture notes: Introduction to Language Processing and Computational Linguistics**, Department of Computer Science, Lund Institute of Technology

**Apache Tomcat of The Apache Jakarta Project**, <http://jakarta.apache.org/tomcat/>

**Amazon.com**, <http://www.amazon.com/>

## Appendix A. Structure of implemented dialogs

This appendix shows the structure of all implemented types of possible and successful dialogs. A "new" dialog (beginning with one of the three main types below) can be started at any point in the current dialog.

Upper case words within brackets indicate information supplied by the user or other dynamic information generated by the program. Below follows a list of descriptions of such words.

|                     |   |
|---------------------|---|
| [TITLE]             | Title supplied by user or found during search                       |
| [AUTHOR]            | Author supplied by user or found during search                      |
| [SECTION]           | First letter in author's last name indicating a section             |
| [HITS]              | Number of hits found during search                                  |
| [DIRECTION]         | Side of the artificial library (left or right)                      |
| [LOCATION_SENTENCE] | A possible sentence of the third main type of dialog starters below |
| [ANYTHING]          | Any sentence/text   |

- Asking for a title (e.g. "What do you have by Stephen King?")

Search result returns no titles:

- I'm sorry, but it seems like we have no books by [AUTHOR].
  - Okay
    - What else can I do for you?

Search result returns just one title:

- I can only find [TITLE] by [AUTHOR]. Is this the one you are looking for?
  - Yes
    - You will find it in the '[SECTION]' section.
      - Okay
        - I am glad I could help you!
  - No **OR** No, [ANYTHING]
    - I'm sorry but it seems like we don't have any other books by [AUTHOR].
      - Okay
        - What else can I do for you?

Search result returns more than one but fewer than six titles:

- I found [HITS] titles:  
[TITLE] ... [TITLE]  
Is it any special book you are interested in?
  - Yes
    - Which one would that be?
      - [TITLE]
        - It would be to your [DIRECTION] in the '[SECTION]' section.
          - Okay
            - I am glad I could help you!
  - Yes, [TITLE] **OR** Yes, [LOCATION\_SENTENCE]
    - It would be to your [DIRECTION] in the '[SECTION]' section.
      - Okay
        - I am glad I could help you!
  - No **OR** No, [ANYTHING]
    - Okay, I hope you find them interesting!

### Search result returns more than five but fewer than twenty six titles:

- I found [HITS] titles. The most popular ones are:  
[TITLE] ... [TITLE]  
Is it any special book you are interested in?
  - Yes
    - Which one would that be?
      - [TITLE]
        - It would be to your [DIRECTION] in the '[SECTION]'
    - Okay
      - I am glad I could help you!
  - Yes, [TITLE] **OR** Yes, [LOCATION\_SENTENCE]
    - It would be to your [DIRECTION] in the '[SECTION]'
  - Okay
    - I am glad I could help you!
- No **OR** No, [ANYTHING]
  - Okay, I hope you find them interesting!

### Search result returns more than twenty five titles:

- I found more than 25 titles. The five most popular, I think, are:  
[TITLE] ... [TITLE]  
Is it any special book you are interested in?
  - Yes
    - Which one would that be?
      - [TITLE]
        - It would be to your [DIRECTION] in the '[SECTION]'
    - Okay
      - I am glad I could help you!
  - Yes, [TITLE] **OR** Yes, [LOCATION\_SENTENCE]
    - It would be to your [DIRECTION] in the '[SECTION]'
  - Okay
    - I am glad I could help you!
- No **OR** No, [ANYTHING]
  - Okay, I hope you find them interesting!

## • Asking for an author (e.g. "Who has written The Hitchhiker's Guide To The Galaxy?")

### Search result returns no titles:

- I'm sorry, but I don't know who the author is of [TITLE].
  - Okay
    - What else can I do for you?

### Search result returns one or more but fewer than twenty six titles:

- I found [HITS] matching titles. Could it perhaps be [AUTHOR]?  
Otherwise, you have to be more specific.
  - Yes
    - Great, I'm glad I could help you!
  - No **OR** No, [ANYTHING]
    - You have to be more specific.  
Otherwise, that is the only author I can find on [TITLE].
      - Okay
        - What else can I do for you?



**Search result returns more than twenty five titles:**

- I found over 25 matching titles. Could it perhaps be [AUTHOR]?  
Otherwise, you have to be more specific.
  - Yes
    - Great, I'm glad I could help you!
  - No **OR** No, [ANYTHING]
    - You have to be more specific.  
Otherwise, that is the only author I can find on [TITLE].
    - Okay
      - What else can I do for you?

• Asking for a location (e.g. "On which shelf can I find The Testament by John Grisham?")

**Search result returns no titles:**

**User supplied both the title and the author in the question:**

- I'm sorry, but it seems like we do not have [TITLE] by [AUTHOR].
- Okay
  - What else can I do for you?

**User only supplied the title in the question:**

- I'm sorry, but I can not find any books named [TITLE].
- Okay
  - What else can I do for you?

**User only supplied the author in the question:**

- I'm sorry, but it seems like we don't have anything by [AUTHOR].
- Okay
  - What else can I do for you?

**Search result returns one or more titles:**

**User supplied both the title and the author or just the author in the question:**

- It would be to your [DIRECTION] in the '[SECTION]' section.
- Okay
  - I am glad I could help you!

**User only supplied the title in the question:**

- If you mean [TITLE] by [AUTHOR], it's to your [DIRECTION] in the '[SECTION]' section. Otherwise, you have to be more specific.
- Okay
  - I am glad I could help you!

## Appendix B. Screenshot

