An Approach to Implement English Language Tutorial System in Natural Language Processing

Raju Barskar\(^a\) and Gulfishan Firdose Ahmed\(^b\)

\(^a\)Assistant Professor, University of Institute Technology, Rajiv Gandhi Pradhyyogiki Vishvidhyalaya, Bhopal (M.P.) Pin-462036, India.

\(^b\)Assistant Professor, Maulan Azad National Institute of Technology, Bhopal (M.P.) Pin-462051, India.

Abstract: - Generation of compound-complex sentences for English tutorial system is a challenging task, because of the Users that interact with this system they make English influent by using this tutorial system and ability of English tutorial system helps users to make growth in any kind of business and organization by taking advantage of English language. The task requires designing of Structural representation, which can encode the information contained in the English sentence, and designing a represent a module, which can be use to learn the English tutorial from the structural representation, with the use of rule sets and the lexicon. We concentrate on implementation of different modules of this system such as the learn phase, test phase, dictionary, pronunciation section which is devoted to generate, in a cognitively transparent way, the right tense for the verb appearing in the exercises presented to the student. This paper is concerned with the specifications and the implementation of a particular concept of word-based lexicon to be used for large natural language processing systems such as machine translation systems. This paper tries to explore the development of English language tutorial system for world. In this model we introduce six phases of this tutorial model and each perform some specific task.

1. Introduction

English Tutoring Systems require three different kinds of knowledge: the domain to be taught, techniques of teaching, and characteristics of the student. As such it is worth considering the extent to which knowledge acquisition techniques for conventional expert systems can be applied, and in particular those which aim to semi-automate the process. In general, tutorial-generating computer systems are designed for use by experienced computer programmers or by individuals who have some knowledge in computer programming. However, there is an increasing number of nonprogrammers, such as language tutors, who would like to generate computer-based tutorials on their own, but they do not have the required technical expertise. Based on the practical needs of such users, we have developed a prototype which enables a non-programmer to generate and to present a computer based tutorial on a personal computer. Additionally, the increase use of computers as an education medium means that it is no longer necessary for tutors and students to be physically in the same place anymore. However, in spite of the advantageous of virtual classrooms, and of the unlimited learning opportunities on the internet, the increasing trend in cross-cultural communications also highlights underlying language barriers. One major problem occurs when both the tutors and the students are monolinguals in their own language; therefore, student-tutor dialogues are impossible. In order to overcome language barriers, the prototype also incorporate an ability to translate student-tutor queries. The main aim here is to enable the students - who are learning the language via computer-based tutorials - to communicate with the tutor without any need to be fluent in each other’s language. English tutorial system uses the concept of natural language processing and natural language understanding.

In this system intelligent feedback module being created by help of NLP and NLU. Natural Language Processing (NLP) is the computerized approach to analyzing text that is based on both a set of theories and a set of technologies. And, being a very active area of research and development, there is not a single agreed-upon definition that would satisfy everyone, but there are some aspects, which would be part of any knowledgeable person’s definition.
Various kinds of languages have been developed for the specification of SICs at different abstraction levels. For example, an ontology-driven language named Semantic Web Rule Language (SWRL) is used by [4] for expressing SICs within ontology. At the implementation level, languages like Spatial SQL [5] are proposed. However, in this paper we focus on the languages used at the conceptual level within the spatial database community, i.e., natural languages [6] [7], a visual language [8], a hybrid natural language [9], spatial OCL [10], spatiotemporal modeling languages [1], and first-order logic [9]. In spite of the capability of spatiotemporal conceptual modeling languages to express SICs directly in the conceptual schema, more specific languages, called Integrity Constraint Specification Languages (ICSL) have been introduced. These languages specify SICs in a data dictionary since they cannot be expressed efficiently in a schema. We classify all these possibilities into four main categories, i.e., natural languages, visual languages, first-order logic language, and hybrid languages.

2. Natural Languages
Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. We classify natural languages into two categories: (1) Free natural language, and (2) Controlled natural language as described below.

2.1 Free Natural Languages
A free natural language is a natural language without implication of any limit on syntax and semantics of the language. Syntax of a language describes the way language’s symbols should be combined to create well-formed sentences. However, semantics reveals the meaning of syntactically correct strings in a language [10]. SICs can be defined using a free natural language. For example, in English: “A road does not cross a building” is the simplest form of specifying SICs. Free natural languages support a rich vocabulary. However, they are sometimes ambiguous or used too loosely. In such languages, several words may bear the same semantics and a word may have several meanings depending on the context. Nevertheless, they remain today’s most widely used language for expressing SICs.

2.1.2 Controlled Natural Languages
Controlled Natural Languages are subsets of natural languages whose grammars and dictionaries have been restricted in order to reduce or eliminate both ambiguity and complexity. Traditionally, controlled natural languages fall into two major categories: those that improve the readability for human readers, in particularly for non-native speakers, and those that improve the computational processing of a text." Traditionally, controlled languages are split into two major categories: (1) CLs that improve human readability, mainly for non-native speakers, and (2) those that constrain the text for computational treatment. The original concept arose during the 1930s, when a number of inertial linguists and scholars devoted considerable effort to establishing a 'minimal' variety of English; It's purpose being to make English accessible to and usable by as many individuals as possible worldwide [18]. For example, in this language, “each road segment should be end-to-end connected to at least one or at most two road segments” is expressed as: Road Segment Touch-Tangent Road Segment [1,2] Where [-,-] is the cardinality of the topological IC indicating minimum and maximum number of instances of the two classes defined in the relationship.

2.2 Visual Languages
A visual language employs graphical and image notations to communicate the information with the user. Various visual languages for different purposes have been proposed; for a survey of these languages refer to [11]. In this section, by visual languages, we mean the languages that use only graphical or image notations. Languages that let users create custom icons and iconic/visual sentences are receiving increased attention as multimedia applications become more prevalent. Visual language systems let the user introduce new icons, and create iconic/visual sentences with different meanings and the ability to exhibit dynamic behavior. Furthermore, visual programming systems support problem solving and software development through the composition of basic software components using spatial operators such as "connect port #1 of component A to port 2 of component B".

2.3 First-Order Logic Language
First-order logic language is a common formal language for representing ICs in databases and knowledge bases. First-order logic allows quantification on objects, i.e., the first-order entities exist in the real world. First-order logic allows reasoning about properties that are shared by many
objects, through the use of variables. For example, let Phil(a) assert that a is a philosopher and let Schol(a) assert that a is a scholar. Then the formula

\[ \text{Phil}(a) \land \neg \text{Schol}(a) \]

asserts that if a is a philosopher then a is a scholar. The symbol \( \rightarrow \) is used to denote a conditional (if/then) statement. The hypothesis lies to the left of the arrow and the conclusion to the right. The truth of this formula depends on which object is denoted by a, and on the interpretations of "Phil" and "Schol".

Assertions of the form "for every a, if a is a philosopher then a is a scholar" require both the use of variables and the use of a quantifier. Again, let Phil(a) assert a is a philosopher and let Schol(a) assert that a is a scholar. Then the first-order sentence

\[ \forall a (\text{Phil}(a) \rightarrow \text{Schol}(a)) \]

asserts that no matter what a represents, if a is a philosopher then a is scholar. Here \( \forall \), the universal quantifier, expresses the idea that the claim in parentheses holds for all choices of a. To show that the claim "If a is a philosopher then a is a scholar" is false, one would show there is some philosopher who is not a scholar. This counterclaim can be expressed with the existential quantifier \( \exists \).

Here:

- \( \neg \) is the negation operator:
  
  \[ \neg \text{Schol}(a) \] is true if and only if

  \[ \neg \text{Schol}(a) \]

  Schol(a) is false, in other words if and only if a is not a scholar.

- \( \land \) is the conjunction operator:
  
  \[ \text{Phil}(a) \land \neg \text{Schol}(a) \]

  asserts that a is a philosopher and also not a scholar.

### 2.4 Hybrid Languages

A number of languages used for the specification of SICs are not purely natural, visual, or logical and can be best described as a combination of them. We call such languages *hybrid languages*. Depending on the dominant component of a language, a hybrid language can be *visual hybrid* or *natural hybrid*.

#### 2.4.1 Visual Hybrid Languages

The main part of visual hybrid languages consists of visual symbols, which are enriched by a natural language. As a limited number of visual constructs are easy to perceive [12], visual hybrid languages combine this advantage with the richness of natural languages. Unified Modeling Language (UML) is such a language. As previously stated, a conceptual model consists of conceptual schema and data dictionary. We did not find any visual hybrid ICSL for defining SICs in data dictionary.

#### 2.4.2 Natural Hybrid Languages

A natural hybrid language consists of a natural language as well as pictograms or symbols, but the natural language part is dominant. **Natural Hybrid Language with Pictograms:** A natural hybrid language with pictograms consists of a natural language which is enriched by pictograms [3]. In the spatial database community, pictograms are intuitive symbols employed for the representation of objects’ geometries. Normand [2] proposes a natural hybrid language with a limited number of pictograms to overcome the limitations of visual hybrid modeling languages in the specification of SICs. This language can specify all types of SICs between object classes, including the SICs that depend on specific attribute values or on objects’ geometries (when alternate or multiple geometries are involved). In this language, class names, topological relationships, and cardinality of relationships are specified by controlled natural language.

### Phases of English Tutorial System

In this section we would like to describe all the modules of English tutorial system that is used to help the users to learn English language. In this system major five modules have been included such as content, intelligent feedback system, dictionary, pronunciation module and collection of database. All the modules prescribe below in detail manner. All the component of this software is well managed to help the user to make English influent explore the key architecture of English language tutorial system.
In this section we would like to describe all the modules of English tutorial system that is used to help the users to learn English language. In this system major five modules have been included such as content, intelligent feedback system, dictionary, pronunciation module and collection of database. All the modules prescribe below in detail manner

2.2 Learning phase

The first stage for many children starting to learn English in the classroom is called the silent stage. In this time they are listening carefully to the language they hear but are not yet ready to start speaking. Depending on the personality and nationality of the child, this stage may last 1 day or 6 months or more. It is important not to be concerned if this stage seems to be taking a long time. The child can learn a great deal without saying a word. In fact, the power of listening is so strong (provided that the language you hear is at the right level of difficulty for you) that one influential researcher has found that one influential researcher into second language learning says that this is all you need to learn a new language! I had evidence of the truth of this theory a few years ago when I taught an Italian boy in grade 6. He refused to say a word of English for more than half the school year, then he suddenly lost his inhibitions and started to talk. Before the end of the same year, he was almost fluent in English and made few serious grammatical errors.

The next stage of language learning comes as students try to acquire control of the English grammar system. Some researchers have found that there seems to be a fixed order in which certain aspects of the system are learned, and this cannot be influenced very much by direct teaching. I see evidence of this every year with my beginner students. They usually get all the answers right. However as soon as they are writing or speaking naturally, without thinking about grammar rules, they forget to use the -s ending. They are just not ready to produce it automatically.

Another typical phase in second language learning is the backslide. By this I mean that students have periods when they get wrong what they seem to have already learned. And they do this in exactly the same way as a child learning English as her first language. Taking English verbs as an example, it very often happens that children seem to have learned some irregular past tense forms. They hear their parents saying sentences like I went to the shops, or I saw a big cow, and they use these correct forms themselves. A little later, however, they start to say sentences like He goed to the shops or I need a big cow. What has happened is that they have started to unconsciously work out the rules of English grammar. They realize that the past tense in English is formed by adding -ed to the verb, but they over generalize this rule out to include all verbs. It takes a little more time before they further refine the rule to include both regular and irregular endings. So this apparent backward step is in fact a sign of progress in learning the language.

Testing Phase

Test the English Query model before compiling it into an application to make sure that the questions users are likely to ask are supported by the model and then modify it accordingly. After you are satisfied with the performance of the model, build it into a compiled English Query application (*.eqd file).

The Suggestion Wizard is available within the Model Test window and is useful during the testing phase. You can start by asking a question that users are likely to ask. Suggestion Wizard presents suggested entities and relationships needed to answer the question.

3.2.1 Model Test Window

The Model Test window in English Query enables you to test the English-to-SQL translation of your model and the execution of the resulting database query before compiling the application.

Note For the data values to appear in the Entity list box in Question Builder, you must select Sample Data on the Data Connections tab before compiling the application.

In the Model Test window (available from the Start command on the Debug menu), you can use the Analysis feature to review the entities for functioning and for questions that return incorrect answers.

When testing a model, questions can be saved to a regression file with the Save Query command. A regression file is automatically created with each project. There can be multiple regression files, but
questions are saved in the first regression file under the project in the Project Explorer window.

An output file can be added and then compared with the original file if the files have names that differ from one another. You can also promote an output file to the new regression file. Regression output is displayed in Extensible Markup Language (XML)-tagged format, which differentiates the questions, restatements, answers, and SQL statements for easier editing.

To maximize writing performance, students should be encouraged:
1. Listen carefully as the testing administrator reads all directions.
2. Read the entire writing prompt carefully and analyze each part for understanding: topic, intended audience, recommended length, type of writing (i.e. letter, essay) and mode of writing (narrative, expository, persuasive).
3. Brainstorm ideas using your favorite graphic organizer: thinking map, four-square, Venn diagram, web, outline, etc.
4. Write rough draft: Stay focused on the assigned topic, present a clear main idea, and support that idea with specific details.
5. Edit your rough draft, paying attention to transitions, flow, vocabulary usage, and organization. The ELA Writer’s Checklist is provided and should be used as a reminder of important aspects of the writing sample.
6. Proof read by checking for correct spelling, punctuation, and grammar. Remember, students are allowed to use both a dictionary and thesaurus for the writing session.
7. When writing your final draft, be as neat as possible. If you make a mistake, simply draw a line through it; it is much neater and quicker than erasing. Students may write in print or cursive. Illegible writing will receive a score of zero.
8. Remember that the writing session will be scored based on the following:
   - Development and support of ideas
   - Expression of ideas
   - Correct sentence formation
   - Usage, mechanics, and spelling

Data Dictionary for English Tutorials System
The prototype has a database which comprises of two parts; one part of the database comprises of the various data collected from all text input by the tutors by the Tutorial Generator. These data are sets of texts for the tutorials; which are referenced individually to the relevant tutor. When a student selects a tutorial in the Tutoring Environment the corresponding data are copied from this part of the database. The second part of the database comprises bilingual examples of English and Malay. The reason for this is that, the Query Translator, uses an example-based approach [5], to avoid the need for complex linguistic analysis and complex computations. However, this approach is highly dependent on a large number of examples in the database. At this stage of system development, a database of examples has been compiled for a demo version of the prototype. There are many bilingual Hindi-English/English-Hindi dictionaries, namely those published by McGraw-Hill, Oxford, Olivetti, Prentice-Hall, etc., as well as several online websites where similar dictionaries and glossaries can be found; for instance, Fernandez Calvo’s computer glossary;1 also Ram-say’s and Lozano-Hemmer’s (1998) “Comparative Cyber Lex-icon”, and Anaya’s (Glossaries de terminus informatics Inglés-Español) or glossaries about common mistakes in translation and language used.. In this system we have included one dictionary that based upon the criteria. Here we are collecting large number of words in favour of forming dictionary.

2.3 Online Tutorial System (English Sentence)
This phase is basically used for online data base system when the internet is connected to system. It is directly gives the answer of the tutorials ask from the student site. You can take a tutorial of the English program’s first unit (Simple Sentences) which is now online connected system. So far four components of the program appear as on-screen icons. The Glossary contains short definitions of grammatical and writing terms. The Tool Kit contains more extensive information on the parts of speech. The software was never meant to teach grammar, but it delivers support information through these resources that are always accessible for just-in-time delivery.

2.4 Communication Phase:
A student's initial interaction with the English Tutorial System is with the Administrator program. The Administrator contains information on all of the students using the system as well as on all of the available instructional texts. Thus new users must
first identify themselves to the system with their user name and then specify their technical area of interest, which may be altered at any point in the future. The available technical areas are maintained in a hierarchical tree structure in order to be able to select a degree of specialization within a given subject area. Individual texts may also reside at more than one node of the tree in order to support inter-disciplinary instruction. After identifying a technical area of interest, subsequent use of the Administrator program will result in a personalized instructional text being selected for the student. Information about the student and the text is then presented to the student on the screen, as is illustrated in Fig. 1. This information includes the identification of the student and the student's stated technical area of interest along with the subject area of the selected instructional text. The subject area of the text may or may not exactly match the technical area specified by the student. For example, if no articles are available in a particular area, or if a student has already mastered all of the articles in their area, the Administrator program will select an article from an area that is as closely related as possible, as defined by the hierarchical tree structure. The texts within an area are all ordered by an objective measure of difficulty based on their vocabulary and grammatical structures, in order to improve the efficiency of the search for an ideal match for the student. A comparison of the contents of the text with the proficiency of the student is presented in the form of three bar charts. These charts identify the percentage of the vocabulary and grammar in the text which the student has either mastered (known area), been exposed to but not fully grasped (review area), or has not seen before (new area). The Administrator attempts to select articles with a relatively large amount of previously mastered material in order to improve comprehension through context and to promote reading rather than dictionary lookup. The Administrator's text selection can be manually overridden by the student either to choose a more basic or more advanced article by simply clicking on the "down level" or "up level" buttons, respectively.

Implementation of English Tutorial Language

A product software implementation method is a systematically structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user. Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. The word deployment is sometimes used Learn More Implementation to mean the same thing.

Implementation phase divided into six sections:

1. **Users/ students**: The first part is to take query from the user in the system.
2. **Learning Phase**: The main strategy introduced in this project was one which empowered students to assume responsibility for their own learning. Thinking things through Learning doesn't happen during lectures or by reading books. It takes place once class is over and your books are closed. Real learning begins when you start tossing what you've read or heard around in your mind, consciously looking for the matches and mismatches between it and what you already know.
3. **Testing Phase**: Tests in the ETS are performance tests. We do not aim at testing the student’s knowledge of English grammar. We would like to test how students produce, understand, and use English in communicative contexts. We are not interested in testing how
well the students remember the facts about English language. We are interested in testing how well the students actually use English in communicative contexts.

4. Data Base Dictionary: The prototype has a database which comprises of two parts; one part of the database comprises of the various data collected from all text input by the tutors by the Tutorial Generator. In this system we have included one dictionary that based upon the criteria. Here we are collecting large number of words in favour of forming dictionary.

5. Online tutorial System: This phase is basically used for online data base system when the internet is connected to system. It is directly gives the answers of the tutorials ask from the student site.

6. Communication Phase: A student's initial interaction with the English Tutorial System is with the Administrator program. The Administrator contains information on all of the students using the system as well as on all of the available instructional texts. Thus new users must first identify themselves to the system with their user name and then specify their technical area of interest, which may be altered at any point in the future.

- Programming:

```vbnet
Private Sub Command1_Click()
    Dim conn1 As New ADODB.Connection
    Dim rs3 As New ADODB.Recordset
    conn1.Open ("provider = Microsoft.ACE.OLEDB.12.0;Persist Security Info=False;Data Source=c:\users\meeta\documents\database 2.accdb")
    rs3.Open "select antonyms from antonyms where word='" & (Text1.Text) & "'", conn1
    a = 1
    Do Until rs3.EOF
        For Each X In rs3.Fields
            Text2.Text = X.Value
        a = rs3.RecordCount
        Next
        rs3.MoveNext
    Loop
    rs3.Close
    conn1.Close
    If a <> -1 Then
        MsgBox ("RECORD NOT FOUND")
    End If
End Sub
```

Results

In this section we recognize the problem and issue that related with English learning process. Students learn the English language with the help of Data Base Dictionary and online tutorial system. After the implementation of English language tutorial system user can interact with the system to take advantage and capability of tutorial system. User can make English influence yourself by using this implemented system as tutorial system. We try to make this system such there is no matter to knowledge of computer. The result will

Conclusion

We have implemented most of the functionalities of the proposed English tutorial system for English. In this paper we have explored the development of English language tutorial system for Indians. In this paper we are defining various components of tutorial system that help user to learn. In this section we would like to describe all the modules of English tutorial system that is used to help the users to learn English language. In this system major five sections have been included such as student or user, learning phase, testing phase, data base dictionary, online tutorial system and communication phase. This paper mainly focus on the learning phase.

References


