ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

AI BASED TUTORING SYSTEM FOR ENGLISH PUNCTUATION: E-VAKYA

Lakshmi Kurup [1], Meera Narvekar [2], Sasikumar M.[3]

[1] Computer Dept., DJSCE, Mumbai, India,

[2] Computer Dept., DJSCE, Mumbai, India

[3] CDAC, Mumbai, India

ABSTRACT

The significance of e-learning technologies has increased rapidly in recent times. With the advent of data from Educational Data Mining applications, public datasets, social media, the design and implementation of intelligent tutoring systems involving adaptive pedagogical strategies and student modelling have become much easier. There are many Intelligent Tutoring Systems (ITS) available in various domains including STEM areas to language learning. In this paper, we present an ITS for learning English Punctuation which provides personalized hints to the students based on the incorrect attempts they make. System brings in automation in presenting the exercises to the students by the use of template structures. The domain knowledge of the system is modeled using syntactic and semantic components of English Language. The knowledge state of the student is captured by the system based on the number of attempts and the time they take to solve an exercise. Pre-test and Post-test analysis has been performed to measure the effectiveness of the system.

Keywords: ITS, ILTS, Domain Module, Tutor Module, t-test, p-value, Cohen's-d

1. INTRODUCTION

According to Oxford Learner's dictionary, the definition of punctuation is given as "the marks used in writing that divide sentences and phrases; the system of using these marks". Punctuation provides meaning and structure to the sentence. Usage of a wrong punctuation or incorrect placement of punctuation can change the meaning of the sentence. So punctuation is an integral part in written English, to guide the reader towards the intended meaning. Proper usage of punctuation has always been a challenge across people of all ages whether native or non-native, as there are many written/unwritten rules. We observe that many grown-ups have problems in this area and end up turning out poorly written documents.

A small pilot study was conducted amongst a class of semester 3 engineering students. A long paragraph was given to them and was asked to punctuate. It was observed that of all the punctuations, people were mostly confused with the usage of commas. Commas were often used when not needed and not used when they were actually needed. Also most of them were unaware of the location of the comma in case of direct quotations. Other punctuations like semicolon and colon were mostly unknown to the majority of the people. Though they were very comfortable with full stop and exclamation, not all the sentences pertaining to these punctuations were correct. The study shows that there is a greater need for teaching punctuation in English Language.

This paper discusses the design of an intelligent tutor that guides the students and presents them with exercises that test a particular concept until the student has mastered that concept. The tutor poses questions, analyses the responses and attempts to offer customized instructions and feedback. When the student finally gets a correct response, the tutor proceeds to the next exercise. The system complements the traditional classroom teaching and can be used for effective personalized tutoring.

A typical architecture of ITS involves four modules: a graphical user interface to take in user inputs and display the hints and score; a domain model to store the syntactic and semantic knowledge; a student model to represent student's knowledge state and the progress he/she makes and finally a tutoring or a pedagogical module which gives personalized instructions to the student and guides him/her to acquire an unknown skill set .Quite a lot of research has been performed on each of these modules till now, dating back to 1970's. In the context of language learning, there are many inherent challenges involved in the design of these modules. Because of the complexities and richness that English allows for a reader/speaker to express themselves in so many ways, the domain knowledge has to be well defined with the help of a subject expert. Detailed analysis and deep understanding of domain knowledge is needed for decision –making process involving generation of hints and feedback. Once the tutoring technologies can dynamically adapt to the specific need of the user, greatest performance gains can be achieved within stipulated time.

This paper outlines our effort to build an adaptive intelligent tutoring approach that teaches the basic rules of English punctuation. We provide a brief overview of our system, E-VAKYA, its domain representation, user modelling and evaluation of data. We start with a brief literature survey in Section II. Section III focuses on the architecture of the system and section IV showcases studies to measure the effectiveness of the system.

2. LITERATURE SURVEY

There are many different aspects to be considered while designing an ITS like the way that the learning and student model is built, the knowledge components of the model, how the knowledge has to be processed and used for guiding the tutoring process. Architecturally, ITS can be categorized as constraint based, model based and dialog based systems. Ohlsson [1] proposed a constraint based model (CBM) where the domain and student model has been characterized by a set of constraints based on the relevance and satisfying conditions. This paradigm is feasible only for domains in which the solution itself is rich in information. The CBM just concentrates on whether the student has reached the goal state irrespective of the number of steps he/she has taken. The model-tracing (MT) tutors [2, 3], the most traditional approach, are cognitive tutors that contain an expert model to trace the student's responses to ensure that the student's responses follow an acceptable solution path. Learner's reasoning skills are evaluated by analyzing the model and so design of an MT tutor takes greater effort than typical CBM models. The most recent advancement in ITS is the use of conversational or dialog based tutors (DBT) [4]that capture the effectiveness of expert human teacher-learner interactions by holding conversations in natural language. The design and implementation of DBT involves significant challenges ranging from domain modeling to natural language understanding/classification and eventually developing a personalized dialog strategy [5]. These issues are magnified when systems are developed at scale and across domains. We intend to go ahead with CBM for our system that teaches the basics of English Punctuation.

Teaching a language or its grammatical constructs is a multi-level task as it requires understanding the meaning and the contextualisation of the vocabulary words [6]. As compared to learning in STEM subjects where the knowledge domain has a fixed set of mathematical theorems or formulae, language learning is quite difficult as the domain consists of thousands of grammatical and lexical rules and also many exceptions to these rules. Though word dictionaries, vocabulary lists and other parts-of-speech datasets are available online, an explicitly well-defined knowledge domain cannot be easily built, as knowledge of each word with respect to its context is not formally defined. Different teaching and assessment strategies must therefore be employed. So classical approaches for designing tutoring services are not feasible for ill-defined domains. To overcome these limitations, a hybrid approach including various AI techniques and methodologies are required for supporting tutoring services. An intelligent Tutoring System in the language domain is called an Intelligent Language Tutoring System (ILTS). ILTSs are systems that employ AI techniques such as Natural Language Processing, Automated Speech Recognition and other machine translation techniques to teach language skills and have the ability to effectively increase the efficiency by automating the learning processes. These AI techniques into ILTS will provide appropriate methods to evaluate students' answers by identifying the text-input errors and provide an immediate feedback to them [6]. There are different ILTSs that have been implemented in different domains in different languages. We now concentrate on the ITSs specific to punctuation tutoring.

A prototype Intelligent Teaching System was designed aimed at improving Dutch university students' use of punctuation in writing and editing texts. In addition to the grammatical aspects of punctuation, the effect of using different punctuation marks with respect to the rhetorical structure of the text was also considered in the system

[7]. The system offers student texts in which he should check the punctuation, and if necessary, make corrections. The system then analyses the student's answer and the differences with respect to possible correct solutions, and gives specific feedback based on these analyses. The system had a drawback that whenever new problems or exercises had to be incorporated, explicit addition of the possible correct solution was needed, even though the rules pertaining to those exercises are the same. A formative evaluation shows that, rather than teaching the rules, the system actually makes students understand how the punctuation works. However, the approach used and performance evaluation is unknown.

A constraint based modelling tutor for learning capitalization and punctuation marks (CAPIT) tried to avoid this bottleneck. CAPIT [7] teaches a subset of the basic rules of English capitalization and punctuation, such as the capitalisation of sentences and the basic uses of commas, periods, apostrophes and quotation marks. The constraints are represented as <Cr, Cs> where Cr is the relevance condition consisting of 25 set of tags like NAME_OF_PERSON, L_CASE etc., and Cs: is a satisfying condition, a regular expression to identify the constraint and hints to be shown when a constraint gets violated. When the user submits a solution, the student model selects the most relevant constraints and violated constraints are passed to the pedagogical module for the

display of error messages. The ITS was designed for school children in the 10-11 year age group. Authors studied how students learned the constraints by taking average of all the violated constraints across nth attempts. The results also showed that the students mastered 25 rules presented in the system effectively. A major observation made was that students hardly asked for detailed explanations. The model was restricted to 45 predefined exercises and 25 sets of constraints. Only one error message was displayed at a time. The rules taught were constrained more on the simpler constructs like capitalisation, possessive nouns etc. with little or no focus on the grammatical and semantic constructs.

Another ITS, a punctuation tutor in Turkish [8], uses CBM and an overlay model for student modeling. Every time a student makes a mistake, feedback and hints are presented. The level of student learning is marked in an overlay model. The overlay model records how close the knowledge level of the student is to that of an expert. This level of learning is determined by the MYCIN certainty factor. A MYCIN certainty value is assigned to each constraint in the database. When a student violates a constraint, negative points are assigned and for each satisfying constraint, positive points are assigned. A nonlinear addition of these points is MYCIN CF. The number of attempts and MYCIN CF are passed as input variables to the fuzzy logic system. The output displays the variables that determine the learning level- Does Not Know, Might Know, Most probably knows. Also the ITS attempts to determine the learning gaps of the student relative to specific topics and concepts.

The Dutch ITS [9] was not adaptable, when new exercises had to be incorporated. Every possible correct solution is explicitly represented for each exercise in the system. This means that when a new exercise has to be added to the database, even if they use the same rules as existing exercises, all their correct solutions must also be added. CAPIT and Turkish based ITS relied on Constraint based modeling which

Constraint number	C_r	C_s	Feedback message
C_1	Start sentence	δ(A–Z)–0)	A sentence should start with a capital letter
C_2	People names	δ(A–Z)–0)	People names should start with a capital letter
C_3	Place names	δ(A–Z)–0)	Place names should start with a capital letter
C_4	Date/month/day names	δ(A–Z)–0)	Month and day names in dates should start with a capital letter
C_5	Lowercase letter	δ(a–z)–0)	The word should start with a lowercase letter

inherently is limited in its focus on learning from errors. Also the constraints defined were on simple rules like capitalization of starting words in a sentence, capitalization of people names, apostrophe words etc. Given below in Fig: 1 represents some of the constraints defined for the Turkish ITS [8]. These constraints are just restricted to a word or location of a punctuation, not the grammatical aspect.

Figure 1: List of Constraints in Turkish ITS

English punctuation has always had two complementary aspects: phonological punctuation linked to how the sentence can be read aloud, particularly to pausing; grammatical punctuation linked to the structure of the sentence. Our ITS intend to teach the punctuations based on the grammatical aspects of the sentence. Therefore the domain model has to be modeled using the structural and grammatical components of the sentence, thereby easing the pedagogical functionalities.

3. DESIGN AND ARCHITECTURE OF E-VAKYA

The paper proposes the design of an Intelligent Tutoring System for learning English Punctuation that provides its potential users with exercises that help them in understanding various key terms and concepts of punctuation. The system should provide appropriate hints/help whenever a student makes a mistake and guide him/her to reach the solution. The errors that students make need to be categorized as erroneous, desirable, undesirable etc. based on which student's performance evaluation would be done. Figure 2 depicts the architecture of our system E-VAKYA.

Figure 2: Architecture of E-VAKYA

The pedagogy relies on the facts: repetition of the exercises of the same level, where each exercise is unique; timely intervention by the tutor and display of appropriate hints based on the attempts made by the student. Most of the ITSs store predefined exercises in the Knowledge Base (KB) which cannot be modified or replaced. To avoid this static nature of domain knowledge acquisition, template structures are used, which automates the sentence generation. The template structure is inspired from the ITS, Marathi E-guru [10]. The pedagogical knowledge base of Marathi e-guru contains templates of various types of Marathi sentences. Domain KB contains vital knowledge of Marathi language which is required in generation of syntactically and semantically valid sentences. In the template structure, the verb is usually fixed. Since the verb is known, we can decide what type of subject and object should be included in the template structure. As per the verb, one can add one or more constituents (adjective, adverb, etc.) and get a new template structure which can be used for generation of similar sentences. Further the template structure can be extended for more complex sentences like imperative, interrogative, exclamatory sentences.

To generate meaningful sentences, sentence generators require correct knowledge of syntax and semantics of the language. The domain model stores the inflection rules in nouns, verbs, adverbs, adjectives and prepositions. Inflections are word elements that indicate grammatical relationships among the words in a sentence. This knowledge is formed from the grammatical rules and the subject-verb-object order that English language follows. This knowledge is represented in a form that is understood directly by the other components of the system. This knowledge representation strongly influences the approach to student modelling and the pedagogical processing of the system. Even though the domain model is not inherently intelligent; the intelligence is achieved by the way the domain model is used throughout the system.

The template structure for a simple sentence "Mary ate an apple" is given below in Figure 3. Each node in the XML structure takes in values from the domain knowledge based on the TYPE parameter. For e.g. The NOUN node gets any person name because the TYPE is "personified". The VERB node gets attached to the word "ate". In the OBJECT node, the TYPE is "eatclass" and so from the domain KB any random names in the eatclass gets attached. Based on the object selected, article "a/an" would be attached depending on the rule of starting vowel of the object. This template structure can be easily extended to add adjective phrases, adverbial phrases, prepositional phrases etc. The system currently has 150 templates customized based on the need of punctuation learning. XML templates for question tags, direct questions, indirect questions, exclamations, interjections, Complex sentences, compound sentences, simple declarative sentences, and interrogative sentences have been designed. Thus XML templates completely automates the generation of input exercises.

```
<SENTENCE_STRUCTURE>
  <NAME>SIMPLE SENTENCE</NAME>
  <SUBJECT>
    <NOUN NAME="x" TYPE="personified"></NOUN>
  </SUBJECT>
  <VERB>
    <VERBMAIN TYPE="ate"></VERBMAIN>
  </VERB>
  <OBJECTPHRASE>
    <ARTICLE TYPE="x"></ARTICLE>
    <OBJECT>
      <OBJNOUN TYPE="eatclass"></OBJNOUN>
    </OBJECT>
  </OBJECTPHRASE>
</SENTENCE_STRUCTURE>
```


Figure 3: A XML template structure for a simple sentence

Each self-generated exercise is tagged with <w> as not desirable error ;< m> as desirable or mandatory error; < O> for optional error and <c> for correct answers.

The sentence generated using the XML template is presented before the student. The student punctuates the sentence .This punctuated text is compared to the tagged sent. Whenever a punctuation appears at <w> or if a punctuation doesn't appear at <m> the error message gets flashed.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Prof		Mary	,	an		old		lady		ate		an		apple	.
Prof	<m>	Mary	<m>	an	<w>	old	<w>	lady	<m>	ate	<w>	an	<w>	apple	<m>

Figure 4: A Comparison of punctuated text with tagged text

In Figure 4, the 2nd and 10th column has no punctuation, when the corresponding tag is mandatory (<m>). Since the domain knowledge base has all the knowledge of the components and the encoded rules, the pedagogical module can easily tailor the presentation of either a generalized hint or a specific hint.

Each punctuation has a different level of exercises. Each level has to be attempted 5 times. Each exercise, guides the students to correct solutions by way of 3-way hints.

1. In the first attempt, if any of the punctuation is not attempted or wrongly punctuated, the system would generate a generalised hint like "Read the manuals properly, incorrect usage of punctuations".

2. The same exercise would be again presented to the student and if the student again makes a wrong attempt, then the exact rules pertaining to the incorrect attempts would be displayed. For the text given in Figure 3, the error would be A title/Degree/ has to be ended with a full stop, when followed by a person's name (column2) , Put a comma after the adjective phrase (column 10).

3. In the third attempt, students would be directly pointed out the location and the appropriate punctuation to be used in that location. For the text given in Figure 3, the messages would be "Put a full stop after the word [Prof]" (column2), "Put a comma after the word [lady]" (column 10).

To generate hints, NLP techniques like POS tagging and NER have been performed using spaCy library [11].

The Student Knowledge base stores the historical data of the student. The student details like name, age, id, score as he/she advances through the exercises along with the time taken to complete each exercise, tags, and hints are stored in the student database. If the student is able to finish the exercise in the first attempt he/she is awarded 5 marks, if he finishes the exercise in the second attempt his marks would be reduced to 60% and if he finishes in the third attempt, the marks would be 40%.

The student knowledge base can be used for further adaptiveness and personalization paving the way to a student model.

4. EVALUATION OF ITS

The ITS was evaluated based on the performance of students both in pretest and posttest. A pretest was conducted on 10 students of different ages from different backgrounds, studying in both Marathi medium and English medium. It was found that irrespective of age, students were comfortable with commonly used punctuations like full stop, comma, question mark and exclamations, but with simple rules. But 50% of students were wrong when it came to lengthier sentences where there were combinations of above said punctuations. 90 % of students attempted other punctuations like semicolon and colon incorrectly. These students used E-VAKYA extensively and finished all the exercises. It was observed that mostly all the students were able to attempt correctly with the third exercise of the same level. After going through each of the exercises (5 Nos for each level), a post test was conducted on the same set of students

A paragraph comprising a mix of punctuations was given to them and it was found that most of the students could perform post-test with utmost accuracy.

Performance is scaled on a range from 0 to 1(based on individual punctuations).The improvement in learning can be verified from gain score analysis [11]. Gain score is calculated by taking out the difference between posttest and pretest scores. A positive gain score indicates that posttest score is greater than pretest and a negative score indicates a less post test score than pretest score. Gain score Analysis is usually done to show the improvement from pre-test to post-test.

Table 1: Min and Max Statistics

	Pre-test	Post test	Gain score (Posttest-pretest)
Minimum	0.05	0.5	0.45
Maximum	0.6	0.9	0.30

The most minimum score in the pretest was 5 and after the posttest, the person who scored a minimum of 5 marks had scored 50 marks (scores are scaled between 0 and 1 in Table1). The positive gain score indicates that the effect of ITS on students is highly significant. Similarly the student who had the highest mark of 60 in the pretest has scored 90 in the posttest, whose gain score of 30 shows the significant improvement of the students after the usage of ITS.

Table 2: Average and Median Statistics

	Pre-test	**Post test**	**Gain score (Posttest- pretest)**
Mean	0.28	0.7	0.42
Median	0.25	0.65	0.40

Table 2 shows mean and median scores. Average score of the students in the pre-test was 28 which further increased to 70 in the post-test. The middle value or the median is 25 in pre-test whereas in post-test the median value increased to 65. As per the pre-test data, we had 5 students who got less than 25 and 5 students above 25 marks, maximum being 50.And as per post-test data, there were 5 students who scored below 65 and 5 students above 65,maximum being 90.

Table 3: Percentile statistics

	Pre-test	**Post test**	**Gain score**
25th percentile	0.0625	0.6	0.54
50th percentile	0.25	0.65	0.4
75th percentile	0.475	0.875	0.4
75th percentile	0.5	0.9	0.4

Percentile is a measure of a student's performance relative to others; it depends on the other students' scores also. In the pretest, a student who scored 6 marks was at the 25th percentile and in the post test, the student is at 25th percentile with 60 marks. Marks scored at each percentile in posttest is much higher than pretest with a minimum of 40 percent gain. All the above descriptive statistics shows that the system has been very effective in teaching the basics of punctuation.

Figure 5: Pre-test and Post-test score

4.1 PAIRED T-TEST

T-test is used to compare the average of the pretest and posttest samples, taken from the same population, at different points of time (before and after the usage of ITS). Student's t-test is performed to determine whether the means of two independent samples are significantly different. Null Hypothesis: $\mu d = 0$ (the mean difference

(d) between both samples is equal to zero) and the Alternate Hypothesis: $\mu d \neq 0$ (the mean difference (d) between both samples is not equal to zero). This is a two-sided test for the null hypothesis that two independent samples have identical averages. Test statistic results obtained is -6.014400 and p-value is 0.000005.Since p-value (=0.000005) < alpha (=0.05), the null hypothesis H0 is rejected, which clearly indicates that because the students have learned the basics of punctuation, their posttest mean score has increased. So we conclude that the students have benefited by the ITS at 0.05 level of significance

Table 2. Two-tailed T-test

Pre-test	**Post-test**	**t (n)**	**p**	**Cohen's-d**
Mean-0.28 SD-0.25	Mean-0.70 SD-0.63	t (10) = -6.014400	0.000005	0.268

The p-values allow us to determine if there is any significant difference between the two tests. But to know how large this difference is, effect size has been calculated using Cohen's – d. In general, a value of 0.2 is considered as a small effect size, 0.5 as medium and 0.8 as a large effect size. For the paired t-test, Cohen's d value is 0.268 (Refer Table 2). Thus, if the means of two groups don't differ by at least 0.2 standard deviations, the difference is trivial, even if the p-value is statistically significant. The conclusion is that a lower d value indicates the need for larger sample size.

5. CONCLUSION

Our system teaches the basic rules of punctuation. The domain model consists of template structures which use semantic KB and grammar rules to generate exercises dynamically. The pedagogical model generates appropriate hints using NLP techniques. The student KB continuously keeps track of the student's progress and stores the score and other performance details. An interesting observation made is that the students of 10-11 years of age performed well as compared to the age group of 13-15 or above. Punctuations like comma, full stop, question mark, quotations, exclamation and hyphen are taught by the system. As a whole, the usage of ITS has shown significant improvement in the learning gains of the students. It has been estimated that the post-test scores are statistically significant than the pre-test scores, which is a clear indication of the usability and effectiveness of the system. The Cohen-d score of 0.27 signifies the need of a large population size. Testing and evaluation has been performed on a small population size, which comprises of 10 students. Increase in population size, would have proven better statistical results. The system was exposed to learners of various ages. It clearly implies that practice and repetition of exercises leads to improved learning. The student knowledge base can be modeled for more personalization. System requires more rigorous testing and validation. Future work of this project involves building of a student model for better personalisation and pedagogical improvements.

REFERENCES

Journal articles

[1] S. Ohlsson, "Learning from performance errors.,," *Psychological Review,* Vols. 103(2),, 1996.

[2] V. Kodaganallur, R. R. Weitz and D. Rosenthal, "A comparison of model-tracing and constraint-based intelligent tutoring paradigms," *International Journal of Artificial Intelligence in Education,* Vols. 15(2),, pp. 117-144, 2005.

[3] V. Aleven, B. McLaren, J. Sewall and K. R. Koedinger, "Example-tracing tutors: A new paradigm for intelligent tutoring systems," Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA, 2008.

[4] S. D'Mello and A. Graesser, "Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to-Human Tutoring," in *Where Humans Meet Machines*, NY, Springer, 2013.

Conference papers

[5] A. C. Graesser, K. VanLehn, C. P. Rose, P. Jordan and D. and Harter, Intelligent tutoring systems with conversational dialogue, 2001.

[6] M. Heilman and M. Eskenazi, "Language Learning: Challenges for Intelligent Tutoring Systems," in *Workshop on Intelligent Tutoring Systems for Ill-Defined Domains, Proc. 8th Int. Conf. Intelligent Tutoring Systems*, 2006.

[7] M. Mayo, A. Mitrovic and J. M. , "CAPIT: an intelligent tutoring system for capitalisation and punctuation," in *Advanced Learning Technologies, 2000. IWALT 2000. Proceedings*, February 2000.

[8] A. Karaci, "Intelligent tutoring system model based on fuzzy logic and constraint-based student model," in *Neural Computing and Applications*, 2019.

[9] A. Mitrovic, K. R. Koedinger and B. Martin, "A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling," in *International Conference on User Modeling*, Berlin, 18 June 2003.

[10] A. R. Joshi and M. Sasikumar, "Approach to Teaching Sentences in Indian Language," in *International Workshop on Technology for Education*, 2009.

[11] E. Kochmar, D. D. Vu, R. Belfer, V. Gupta, I. Vlad and J. Pineau, "Automated Personalized Feedback Improves Learning Gains in An Intelligent Tutoring System," in *Artificial Intelligence in Education*, Cham, Springer International Publishing, 2020, pp. 140-146.