IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

A Light-Weight Automated Approach To Classify and Annotate English Sentences

Subham Sahu[1], Jitendra Singh Thakur[2]

Department of Computer Science and Engineering, Jabalpur Engineering College, Jabalpur, India[1,2]

Abstract: Natural languages like English, on the one hand, offer a lot of flexibility to express a piece of information in a variety of ways, but, on the other hand, pose challenges for automatic information extraction as the relevant information is embedded at different places in different types of sentences. This necessitate the NLP applications to first identify the sentence types so that the relevant information can be extracted from them accordingly. The training, testing and evaluation of such applications need large corpora annotated with the sentence types; however, the available open/free corpora are not annotated with sentence types. In this paper, we propose an automated approach along with a prototype tool support (AutoSA) to classify the English sentences into four types (Declarative, Imperative, Interrogative and Exclamatory), and to classify each type of the sentence further into the four subtypes (simple, compound, complex and compound-complex). First, the approach uses the Stanford Natural Language parser API to generate parse trees of the sentences. Then, it scans the parse tree of each sentence to search for the patterns that we have framed for the classification. Based on the pattern (s) found in the parse tree of the sentence, the approach annotates the sentence with the corresponding type and subtype. We present the two case studies that we have conducted for the validation of the proposed approach. The results of the case study conducted on 2182 sentences of Tatoeba Corpus showed 98.6% precision and 99.5% recall, and the results of the case study conducted on 1650 sentences of Penn Treebank Corpus showed 92.4% precision and 99.4% recall.

1 INTRODUCTION

Natural languages are accomplished with varieties of sentence types and subtypes [1–4]. These variety of sentence types and subtypes facilitate us with an ability to express a similar concept in many ways; however, they pose challenges in automatic information extraction as the relevant information is embedded at different places in different types of sentences [5–7].

The information extraction approaches which identifies domain models from textual specifications such as [5–10] achieve good results by having separate algorithms to handle different types of sentences. [11, 12] observed that the performance of language modeling3 can be improved if the sentences are modeled on the basis of their types where the types can be based on topic, style, or some other criteria. An intelligent automated conversation agent, in order to provide proper responses to its user queries, first, requires identifying the class or type of the input sentences [13].

On the basis of the functionality represented by the sentences, they can be classified into four types: Declarative Sentences, Imperative Sentences, Interrogative Sentences and Exclamatory Sentences [1, 2, 4]. On the basis of the types of clauses present in them, each type of sentences can further be classified into four subtypes: Simple Sentence, Compound Sentence, Complex Sentence and Compound-Complex Sentence [1–3]. An automated approach that can systematically classify the sentences into these types and subtypes can help in improving the performance of many NLP applications such as Information Extraction, Sentence Simplification, Text Summarization, Question-Answering and Machine Translation. The relevant information can be correctly extracted from the sentences by the NLP applications if they first classify the sentences. Once the type and subtype of the given sentences are identified, they can identify their macro and micro structures to correctly extract the required elements from different places in the sentences. By macro structure of a sentence, we mean the broad structure of the sentence depicted by the arrangement of clause(s) in the sentence. By micro structure, we mean the arrangements of finer level semantic elements such as subject, verb, direct object, indirect object and preposition object present in the clause(s) of the sentence.

With the help of two declarative complex sentences (sentences of type ‘declarative’ and subtype ‘complex’), the following example shows how the macro and micro structures of the sentences can be identified.

3The art of determining the probability of words sequences in a sentence

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Example 1. Sentence 1: "When the tank is filled, the system stops the motor." has macro structure SubordinateClause-MainClause
Sentence 2: "The system stops the motor when the tank is filled." has macro structure MainClause-SubordinateClause

Sentence 1 and Sentence 2 both are declarative complex sentence conveying same information (their subordinate clause represents a condition and the main clause represents an action to be carried out if the said condition is fulfilled) but have different macro structures. Having identified the macro structures of these sentences, an NLP application can extract the subordinate clause and main clause from the sentences and then can identify the micro structures of each of these clauses.

Here, the subordinate clause "When the tank is filled" has micro structure "Adverb-Subject-AuxVerb-PassiveVerb" which represents a condition.
The main clause "The system stops the motor" has micro structure "Subject-Verb-DirectObject" which represents an action (i.e., the subject "system" performs the action "stop" on the direct object "motor").
Table 1 shows some possible macro structures of Declarative type sentences. The approach for identifying the micro structures of the sentences is presented in our previous work [6, 7] in which the micro structures of the sentences were named as sentence structures.

Table 1: Some possible macro structures of declarative sentences

Sentence type	Sentence subtype	Some possible macro structures	Example sentences
Declarative	Simple	Clause	The user enters the cash amount.
Declarative	Compound	Clause-Conjunction-Clause,	The balance is deducted from the account, but the cash is not dispensed.
Declarative	Complex	Clause-Semicolon-Clause Subordinate Clause-Main Clause, Main Clause-Subordinate Clause, MainClausePart1-SuboridnateClause-MainClausePart2	The cash despenser dispenses the cash amount; the receipt printer prints the re-ceipt. When the tank is filled, the system stops the motor. The system stops the motor when the tank is filled. The mobile that I want is available online.

A number of approaches for sentence classification have been proposed in the literature (Table 2). The approach of [14] classifies the sentences as facts or opinion, others classify them as subjective or objective [15, 16]. The approaches like [17–19] classify them as well formed or ill formed. [20] tags the sentences with various speech acts. The approach of [21] classifies the sentences as requests or others. The approaches [22–26] classify the sentences in Email conversations into Questions and others. [13] classifies the sentences into Declarative, Imperative and Interrogative.

In this paper, we propose an automated approach along with a GUI-based prototype tool Automated Sentence Analyser (AutoSA)4 to classify the English sentences into the four types (Declarative, Imperative, Interrogative and Exclamatory), and to classify each type of the sentence further into the four subtypes (simple, compound, complex and compound-complex). To accomplish this task, the approach relies on the parse trees of the input sentences obtained with the help of the Stanford NL parser. We identified a set of parse tree patterns to classify the sentences into the four types (Section 2.2.1) and a set of parse tree patterns to further classify them into the four subtypes (Section 2.2.2). The approach first reads the sentences from a corpus then using the Stanford NL Parser API5 generates the parse trees of the sentences. It then scans the parse tree of each sentence to search for the patterns. Based on the patterns found in the parse tree of the sentence, the approach annotates the sentence with the identified type and subtype, respectively. We conducted two case case studies in which we used AutoSA to classify and annotate the sentences taken from two corpora: i) Tatoeba Project Corpus, ii) Penn Tree Bank Corpus. The results showed 98.6% precision and 99.5% recall for Tatoeba Corpus sentences, and showed 92.4% precision and 99.4% recall for Penn Treebank Corpus sentences.
The presented work is the outcome of our experience in the domain of Information Extraction, specifically, the generation of domain models from textual specifications (i.e., software requirements specifications) [5–8,10].

4http://serg.iiitdmj.ac.in/tools/AutoSA (accessed October 7, 2016)
5http://nlp.stanford.edu/software /lex-parser.shtml (accessed March 27, 2016)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 2: Different approaches for sentence classification

Approach	Input	Classification Output (Classes or types of sentences)
[14]	Text document	1) Fact, opinion or uncertain; 2) opinion labeled as positive, negative, neutral, mixed or uncertain
[16]	Text document	Subjective and Objective
[17]	Text document	Erroneous and Correct
[18]	Text document	Erroneous and Correct
[19]	Text document	Grammatical and Ungrammatical
[20]	Email and Online Forum conversations	Speech act tags: Accept response, Acknowledge and appreciate, Action motivator, Polite mechanism, Rhetorical question, Open-ended question, Or-clause question, Wh-question, Yes-no question, Reject response, Statement, Uncertain response
[21]	Email conversations	Request and other
[22]	Email conversation	Question and other
[23]	Online forum thread	Question and other
[24]	CQA	Question and other
[25]	Twitter Micro-text	Question and other
[26]	Email conversation	1) Question and other; 2) Question as Interrogative, Declarative and Imperative
[13]	Text document Text	Declarative, Imperative and Interrogative
Our approach	document	1) Types: Declarative, Imperative, Interrogative and Exclamatory; 2) Subtypes: Simple, Compound, Complex and Compound-Complex

FAQ = Frequently asked questions pages, CQA = Community based Question-Answer Services

The relevant information to be extracted (the problem level objects and the interactions between them; the domain classes, their attributes, operations and the relationships between the classes) from the textual specifications are embedded at different places in different sentences. Based on the different types, subtypes of the sentences and their structures, we framed a set of extraction rules to extract the relevant elements from the textual specifications. For framing the information extraction rules (i.e. Transformation Rules), and the training and testing of the prototype tools6 of these approaches, the corpora of the sentences annotated with sentence types and subtypes were required which we obtained with the help of the tool AutoSA7 presented in this paper.

Specifically, our contributions in this paper are as follows:

1. We propose an automated approach to classify English sentences with their types and subtypes
2. We present the prototype tool, Automated Sentence Analyser (AutoSA)
3. We present two case studies for the validation of the approach
4. We also present the two corpora of English sentences: i) Tatoeba Project Corpus and ii) a part of the Penn Tree Bank Corpus annotated using AutoSA with types and subtypes of the sentences

The rest of the paper is organised as follows. The proposed approach is presented in Section 2. Section 3 presents the two case studies that we have conducted for the validation of the proposed approach. The benefits and limitations of the proposed approach are discussed in Section 4. Section 5 presents the tool AutoSA in which we have prototyped the proposed approach. Section 6 presents the related work. Section 7 presents the conclusion and future directions of the proposed approach.

2 PROPOSED APPROACH

The proposed approach uses parse trees (Section 2.1) of the sentences generated using the Stanford NL parser. With the help of various drilling experiments, we analysed the parse trees generated by the parser for various sentences of each type (Declarative, Imperative, Interrogative, and Exclamatory) and subtype (Simple, Compound, Complex, and Compound Complex) collected from sources such as English grammar books, research papers and websites, and identified a set of subtree patterns (Section 2.2) that can be used to classify the sentences into the mentioned types and subtypes. The working of the approach is described in Section 2.3.

6http://serg.iiitdmj.ac.in/tools/AutoSDG, http://serg.iiitdmj.ac.in/tools/AnModeler (accessed October 7, 2016)
7http://serg.iiitdmj.ac.in/tools/AutoSA (accessed October 7, 2016)

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 11, November 2021
DOI: 10.17148/IJARCCE.2021.101132

2.1 ParseTree

A parse tree (or Phrase Structure Tree) represents syntactic structure (organisation of various clauses and/or phrases) of a sentence. Before parsing the sentence, the NL parsers first tokenize the sentence (divide the sentence into tokens such as words and punctuations). These words or tokens are present in the terminal nodes (or leaf nodes) of the parse tree of the sentence generated by the parser. The root of the parse tree is labeled as ROOT, the internal nodes (or subtrees) are labeled with syntactic tags (or node labels) which represent the clauses and the phrases in the sentence. The pre-terminal nodes of the parse tree are labeled with parts of speech tags (POS-tags). A few such node labels and sentence elements represented by them along with example sentences and parse trees of the sentences are shown in Table 3. For the complete set of the node labels, please refer [27, 28].

2.2 Classification patterns

This section presents the parse tree patterns that we have identified for classification of sentences: i) based on functionality of the sentences, ii) based on types of clauses present in the sentences.

2.2.1 Sentence Type Patterns

On the basis of functionality of sentences, the sentences can be classified into four types [1, 2, 4]:

i) Declarative Sentences (also called Assertive sentence or Statements): These sentences are used to make some statement. These are the most common sentences. In these sentences subjects come first, followed by predicates. These sentences ends with a period (.)
ii) Imperative Sentences: These sentences are used to give orders or to make requests. These sentences normally do not contain subjects and their verb is in the base form. These sentences also ends with a period (.)
iii) Interrogative Sentences: These sentences are used to ask questions. These sentences ends with a question mark (?)
iv) Exclamatory Sentences: These sentences are used to express emotions and strong feeling. These sentences ends with an exclamation mark (!)

With the help of various drilling experiments, we analysed various sentences of the above types (more than 200 sentences of each type collected from sources such as English grammar books, research papers and websites) and their parse trees generated using the Stanford NL parser, and identified a set of subtree patterns that can be used to classify the sentences into the above mentioned four types. We call these patterns as Sentence Type Patterns (Table 4).

2.2.2 Sentence Subtype Patterns

On the basis of types of clauses present in sentences, they can be classified into four types: Simple Sentences, Compound Sentences, Complex Sentences and Compound-Complex Sentences [1–3]. A clause is a group of words that consists of a subject and a predicate; it can be of two types: i) Independent clause (or Main clause) which expresses a whole event or situation in its own , ii) Dependent clause (or Subordinate clause) which requires an another clause (Independent Clause) to express a whole event or situation.

1) Simple Sentence: A sentence that contains only one independent clause. Examples:
i) "The ATM Customer enters the withdrawal amount."
ii) "The system commands the motor to start."
iii) "The system validates the record entered by the customer."
iv) "The ATM card is ejected by the system."

2) Compound Sentence: A sentence that contains two or more independent clauses which are joined using coordinating conjunctions or semi colon (;).
Examples:
i) "The system prompts the user to enter the PIN; the user enters the PIN."
ii) "The Card reader ejects the ATM card, and the printer prints the receipt."
iii) "The system debits the customer's account, but the cash is not dispensed."

3) Complex Sentence: A sentence that contains an independent clause and one or more dependent clauses which are usually joined by the subordinating conjunctions or relative pronouns.

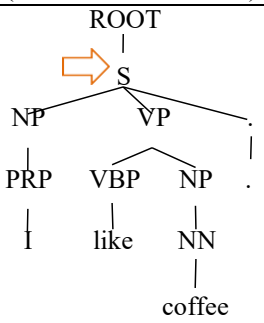
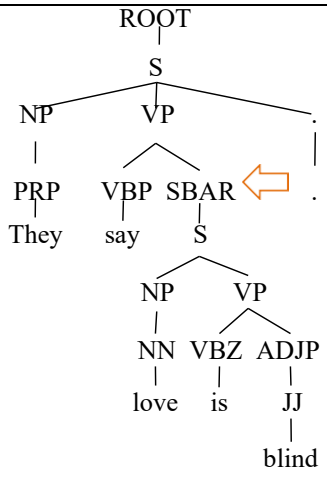
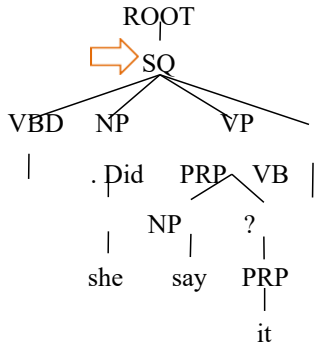
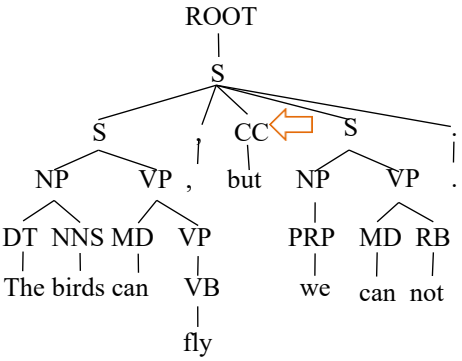
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 3: Some node labels or syntactic tags of parse trees

Node label	Sentence elements represented	Example sentence	Parse tree showing the node label (marked with an arrow)
S	A simple declarative clause or a clause which does not start with a (possible empty) subordinating conjunction or a Wh-word and which not exhibit subject-verb inversion.	I like coffee.	
SBAR	A subordinate clause or a clause which starts with a (possibly empty) subordinating conjunction.	They say love is blind.	
SQ	A direct Yes/No question	Did she say it?	
CC	A coordinating conjunction	The birds can fly, but we can not.	

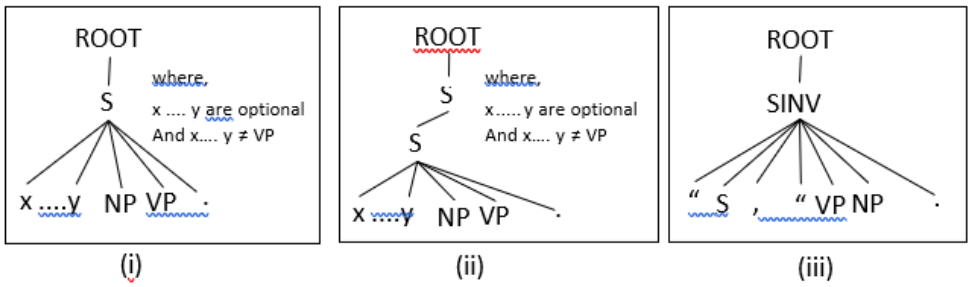
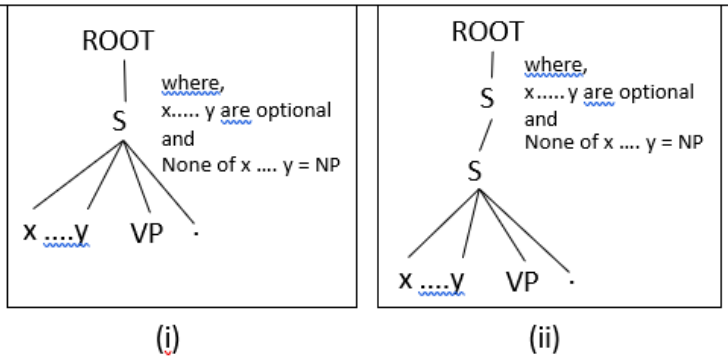
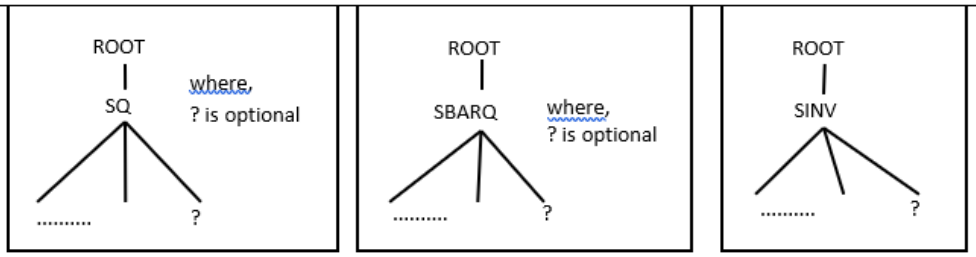
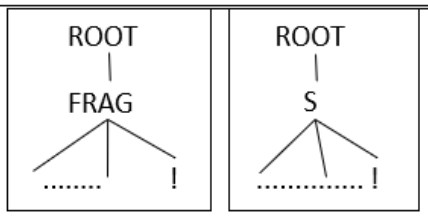
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 4: Sentence Type Patterns

Sentence type	Subtree patterns
Declarative	
Imperative	
Interrogative	
Exclamatory	

S: A simple declarative clause, SBARQ: Direct question introduced by wh-element, SQ: Yes/no questions and subconstituent of SBARQ excluding wh-element,
SINV: Declarative sentence with subject-aux inversion, NP: Noun phrase, VP: Verb phrase, FRAG: Fragment

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Examples:
i) *"The system checks that the card is valid."*
ii) *"While the cash dispenser is counting the cash, the card reader ejects the card."*
iii) *"The ATM card which the customer inserted was invalid."*

4) ***Compound-Complex Sentence:*** A sentence which is the combination of both Compound Sentence(s) and Complex Sentence(s) is called Compound-Complex Sentence
Examples:
5) *"The Customer inserts the card, and the system checks that the card is valid."*
6) *"While the cash dispenser is counting the cash, the card reader ejects the card, and the system starts displaying the message to collect the card and cash."*
For framing the patterns to identify the above mentioned types of the sentences, we analysed various sentences of the above types (collected from sources such as English grammar books, research papers and websites) and their parse trees generated using the Stanford NL parser, and identified a set of subtree patterns named as Sentence Subtype Patterns (Table 5).

2.3 Working of the approach

The proposed approach identifies the types and the subtypes of the input sentences in three steps (Figure 1). First, the approach reads each sentence from the text document, and generates parse tree of the sentence (Step 1). Then, it searches for Sentence Type Patterns in the parse tree of the sentence, and based on the pattern found, it annotates the sentence with the corresponding sentence type (Step 2). Finally, it searches for Sentence Subtype Patterns in the parse tree, and based on the pattern found, it annotates the sentence with the corresponding sentence subtype (Step 3).

2.3.1 Read and Parse sentences

This step reads in sequence the sentences from the text document, and parses each sentence using the Stanford NL Parser API to generate its parse tree.

2.3.2 Classify and annotate the sentences based on functionality of sentences

In this step, the approach searches for Sentence Type Patterns (Table 4) in the parse tree of the sentence, and based on the pattern found in the parse tree, it annotates the sentence with the corresponding sentence type using the annotation character shown in Table 6. If any of the Sentence Type Patterns is not found then it annotates the sentence as U (to represent Unidentified sentence type).

2.3.3 Classify and annotate the sentences based on types of clauses

This step applies the following tests for compound, complex, compound-complex and simple sentences on the parse tree of each sentence to annotate the sentence with the corresponding subtype using the annotation character shown in Table 6. If the sentence does not qualify any of these tests then it is annotated as U (Unidentified).

i) Test for Compound Sentences A sentence qualifies the test for a compound sentence if its parse tree contains at least one of the compound sub-tree patterns shown in Table 4. A sentence that qualifies the test for the compound sentence and does not qualify the test for the complex sentence is annotated as C (Compound Sentence).
Example 2. The parse tree of the sentence "The system dispenses the cash amount, and the card reader ejects the card." shown in Figure 2 contains the compound sentence pattern number (i), and does not contain any complex sentence patterns (Table 4), therefore the sentence qualifies the test for compound sentence.

ii) Test for Complex Sentences To identify complex sentences the approach checks that the input sentence contains a subordinate clause or not. A sentence qualifies the test for the complex sentence if its parse tree contains at least one of the complex sentence patterns shown in Table 4. A sentence that qualifies the test for complex sentence and does not qualify the test for compound sentence is annotated as CX (Complex Sentence).

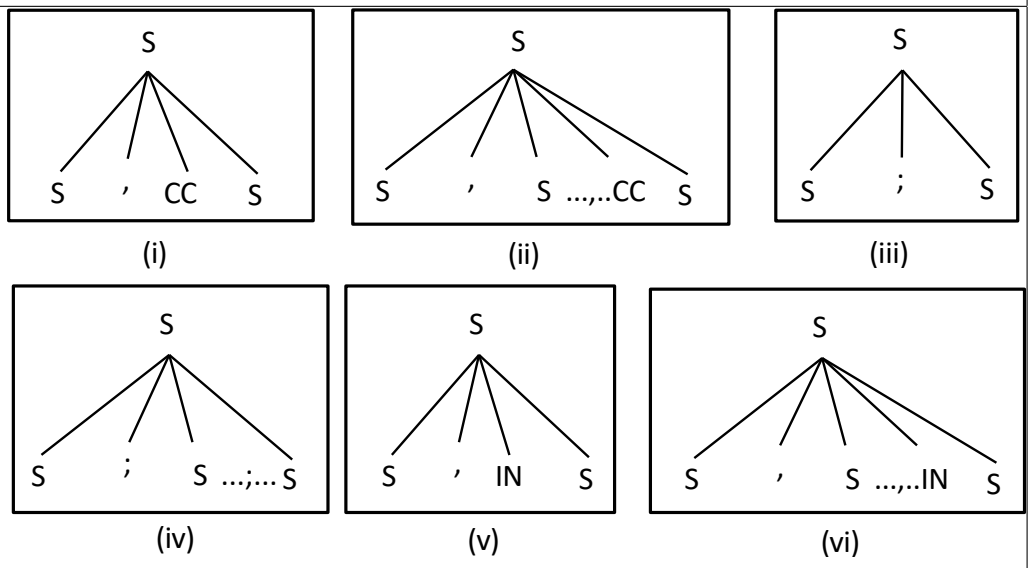
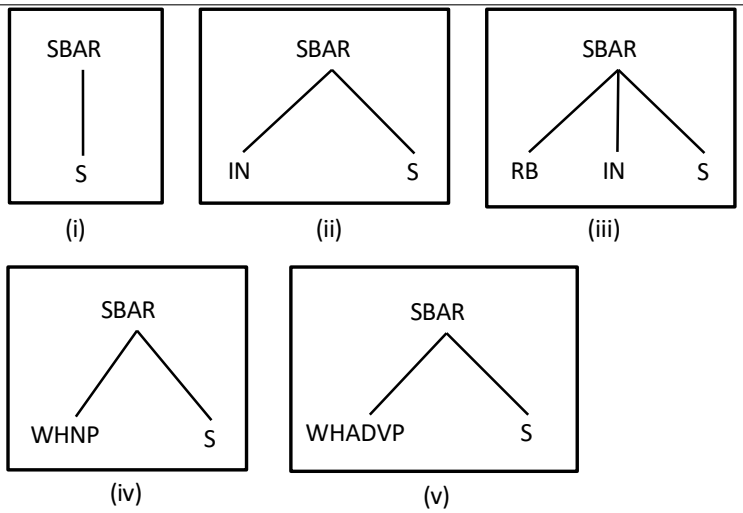
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 5: Sentence Subtype Patterns

Sentence sub-type	Subtree patterns to identify sentence subtypes
Compound	
Complex	
Compound-Complex	A sentence which contains at least one Compound pattern and at least one Complex pattern is identified as Compound-Complex Sentence
Simple	A sentence which does not contain any of the Compound and Complex patterns is identified as Simple Sentence

S: A simple declarative clause, SBAR: Subordinate clause,
WHADVP: Wh-adverb phrase, WHNP: Wh-noun phrase, IN: Preposition,
CC: Coordinating conjunction

Table 6: Annotation characters

Sentence type	Annotation character	Sentence subtype	Annotation character(s)
Declarative	D	Simple	S
Imperative	I	Compound	C
Interrogative	Q	Complex	CX
Exclamatory	E	CompoundComplex	CCX
Unidentified	U	Unidentified	U

IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

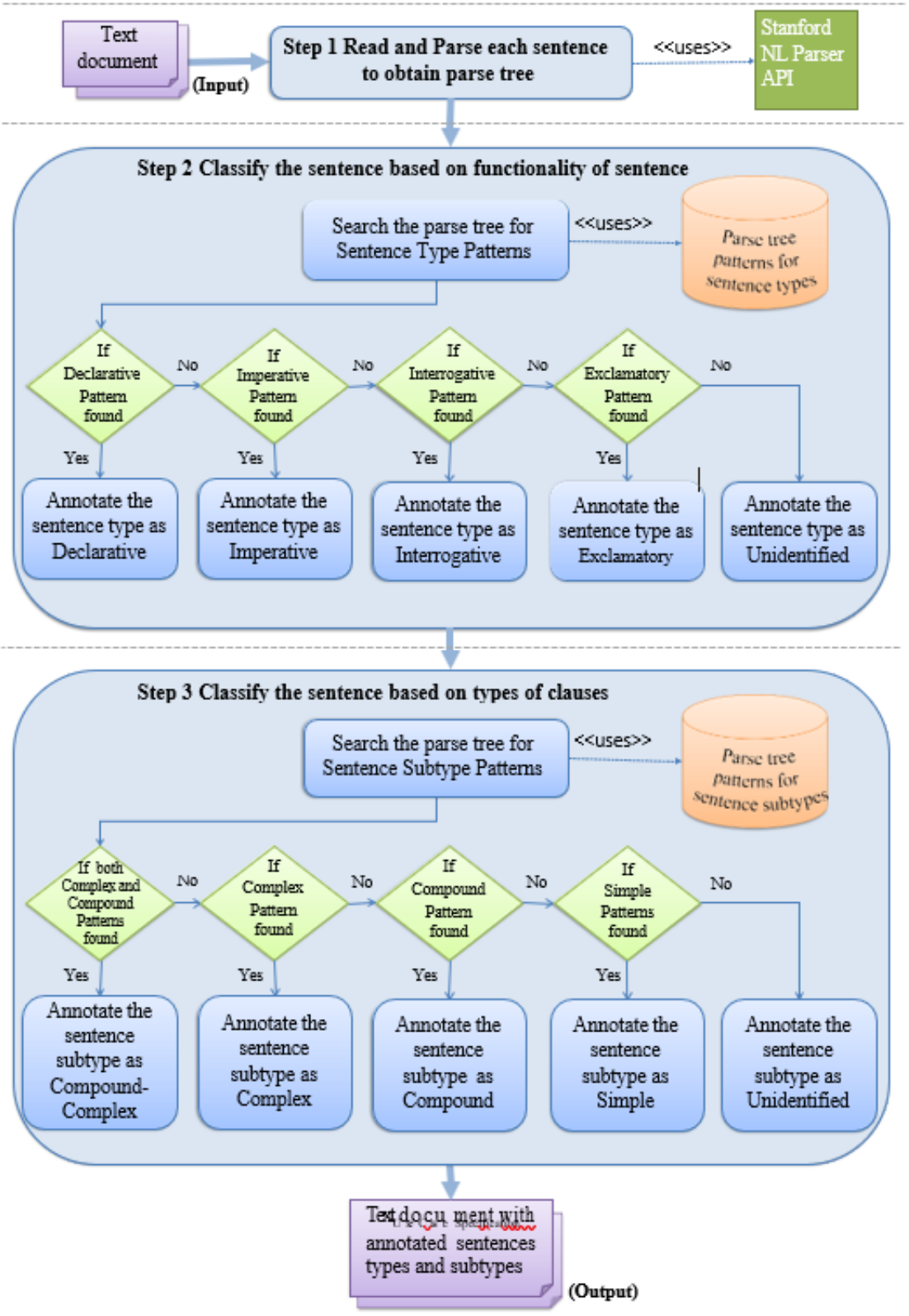


Figure 1: Schematic diagram of proposed approach

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

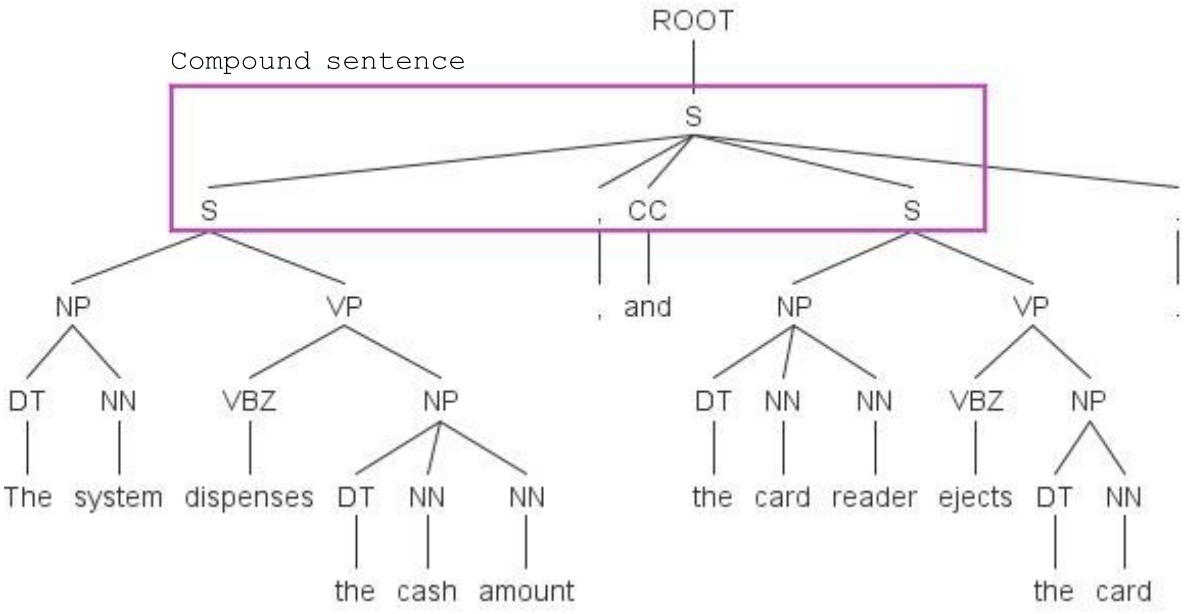


Figure 2: Parse tree with Compound Pattern marked

Example 3. The parse tree of the sentence "If the username and password are correct, the system displays the homepage." shown in Figure 3 contains the complex sentence pattern number (ii), and does not contain any compound sentence patterns (Table 4), therefore the sentence qualifies the test for the complex sentence.

iii) Test for Compound-Complex Sentences A sentence that qualifies the test for compound sentence as well as the test for complex sentence is annotated as CCX (Compound-Complex Sentence).

Example 4. The parse tree of the sentence "If the ATM card is valid, and the PIN is correct, the system displays the transaction options." shown in Figure 4 contains the compound sentence pattern number (i), and also contains the complex sentence pattern number (ii) (Table 4), therefore the sentence qualifies the test for Compound-Complex sentence.

iv) Test for Simple Sentences A sentence that does not qualify the tests for both the compound sentence and the complex sentence is annotated as S (Simple Sentence)

Example 5. The sentence "The customer selects the withdrawal option." does not qualify the test for compound sentences and the test complex sentences as its parse tree shown in Figure 5 does not contain any of the compound as well as complex patterns (Table 4). Therefore the sentence is identified as Simple Sentence.

3 CASE STUDIES

This section presents the evaluation studies that we have conducted to evaluate the precisions and recalls of the proposed approach in identifying the types and subtypes of the sentences.

3.1 Case Study 1 : Tatoeba Project Corpus

The objective of this case study is to compare the accuracy of the proposed approach with the accuracy of human annotators in identifying types and subtypes of the sentences collected from the Tatoeba Project Corpus[8]. Tatoeba Project Corpus is a very large collection of sentences and their translations in different languages available for free use under Creative Common License[9]. These sentences are the contributions of various people across the world collected through a web based interface provided by the Tatoeba Project.

[8]https://tatoeba.org/eng/downloads (accessed March 27, 2016)

[9]http://creativecommons.org/licenses/by/4.0/ (accessed March 27, 2016)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

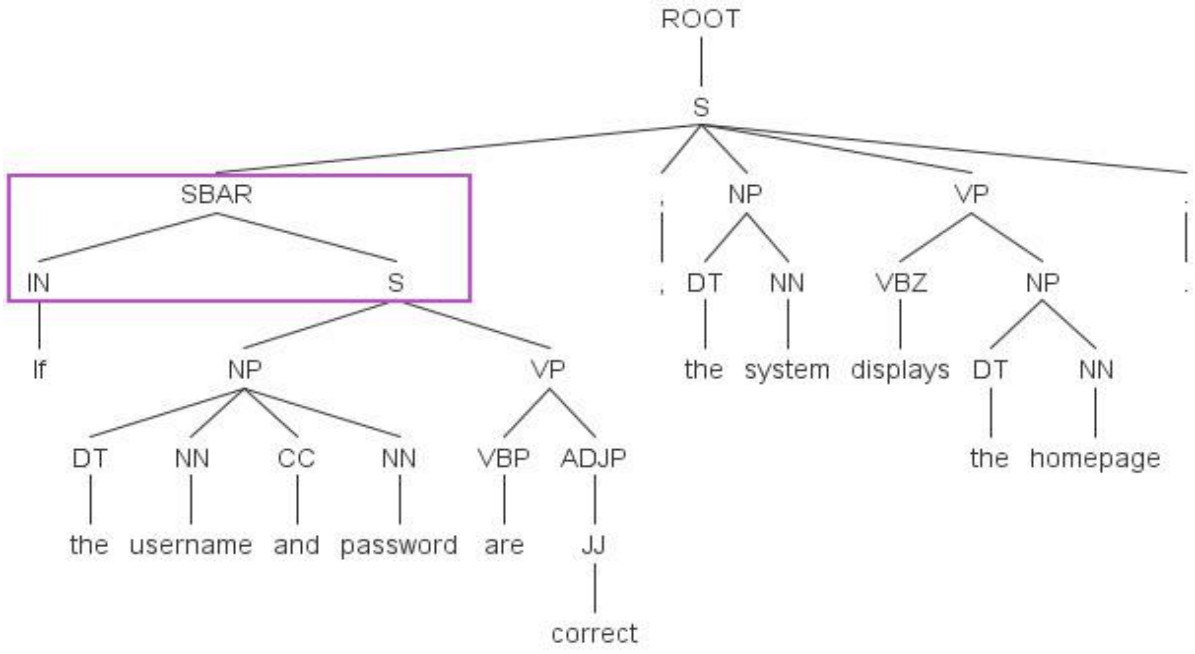


Figure 3: Parse tree with Complex Pattern marked

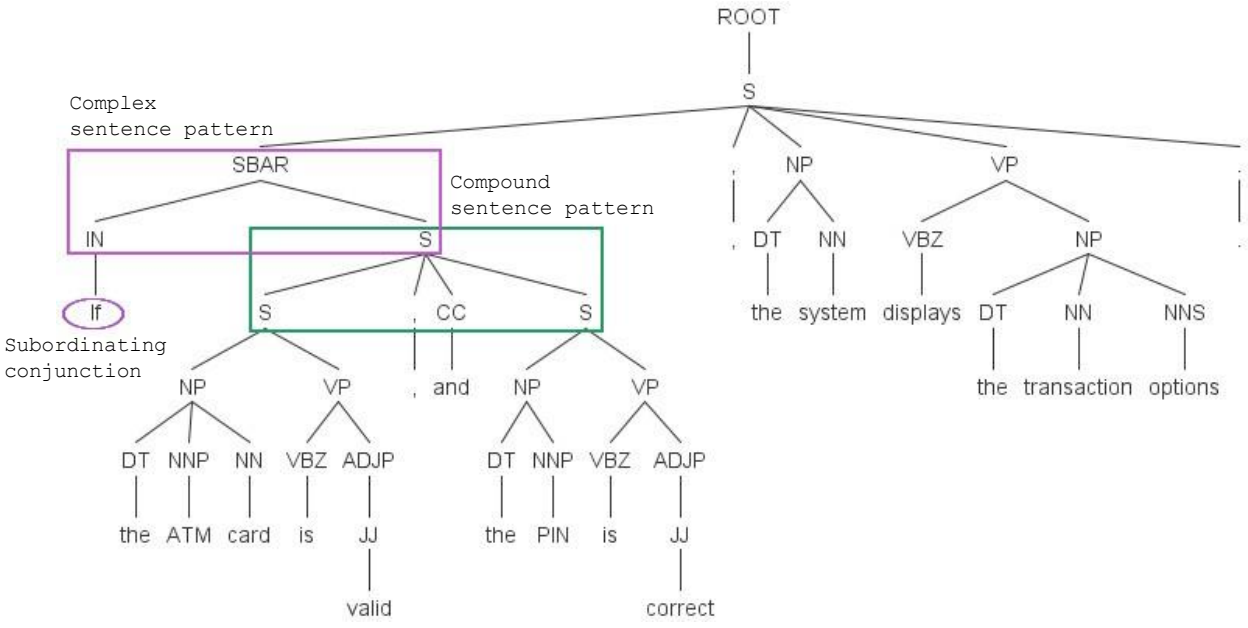


Figure 4: Parse tree with Compound and Complex Patterns marked Simple sentence pattern

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

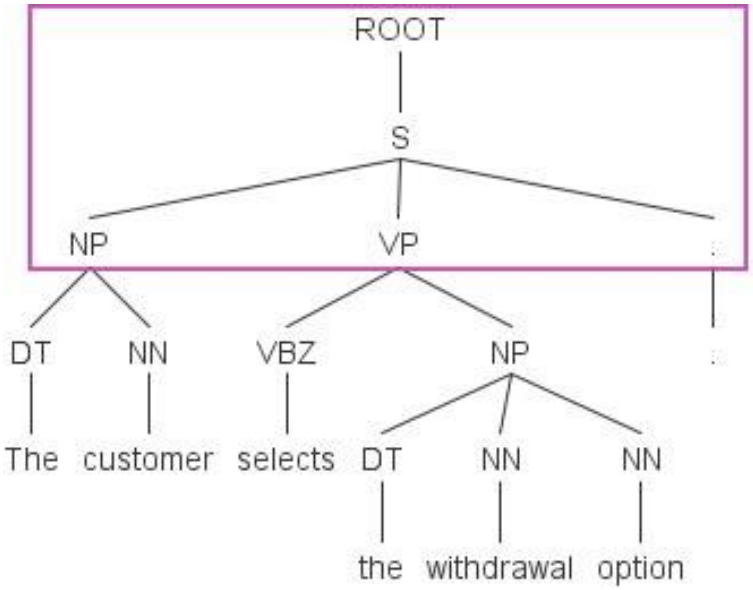


Figure 5: Parse tree with Simple Pattern marked

sentences are stored in a text file in the CSV (Comma Separated Values) format, in which the first value is the sentence ID, the second value is the language ID (that represents the language in which the sentence is written for e.g. the language ID "eng" represents the the sentence is an English sentence) and the third value is the sentence itself.

3.1.1 Design of the study

As the sentences annotated with their types (Declarative, Imperative, Interrogative or Exclamatory) and subtypes (Simple, Compound, Complex or CompoundComplex) were not available, we randomly extracted 2400 English sentences from Tatoeba Project Corpus, and stored them in an Excel sheet. These sentences need to be annotated with their types and subtypes by a human annotator as well as *AutoSA* to evaluate the precisions and recalls.

The independent variables were the types and subtypes of the sentences identified by the tool, and the dependent variables were the precision (P) and recall (R) with which the types and subtypes were identified by the tool.

3.1.2 Execution of the study Manual annotation of the sentences:

The first author thoroughly examined each sentence and annotated it with its type (Declarative, Imperative, Interrogative or Exclamatory) and subtype (Simple, Compound, Complex or CompoundComplex) using the annotation characters shown in Table 6. While manually annotating the sentences, we found that some sentences (218 out of 2400) were incorrectly formed which were removed to obtain 2182 sentences annotated with their types and subtypes. The manual annotation process took about 60 hours time (i.e. two hours per day for 1 month).

The annotations were checked by the other two authors of the paper to evaluate the precision and recall of the annotations done by the first author.

Automated annotation of the sentences:

The 2182 sentences were given as input to *AutoSA* (a tool support implemented based on the proposed methodology for identifying the types and subtypes of the sentences, presented in Section 5). The output (i.e. the sentences annotated with their types and subtypes) was exported by *AutoSA* in an Excel sheet. The time required in automatic annotation was 20 minutes (When AutoSA was executed on a computer with Intel Core i5 2.6 GHz CPU, 4GB RAM and 32 bit Windows 8 OS). The annotations done by the tool were then compared to those done by the human annotator to evaluate the precision and recall of the tool.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

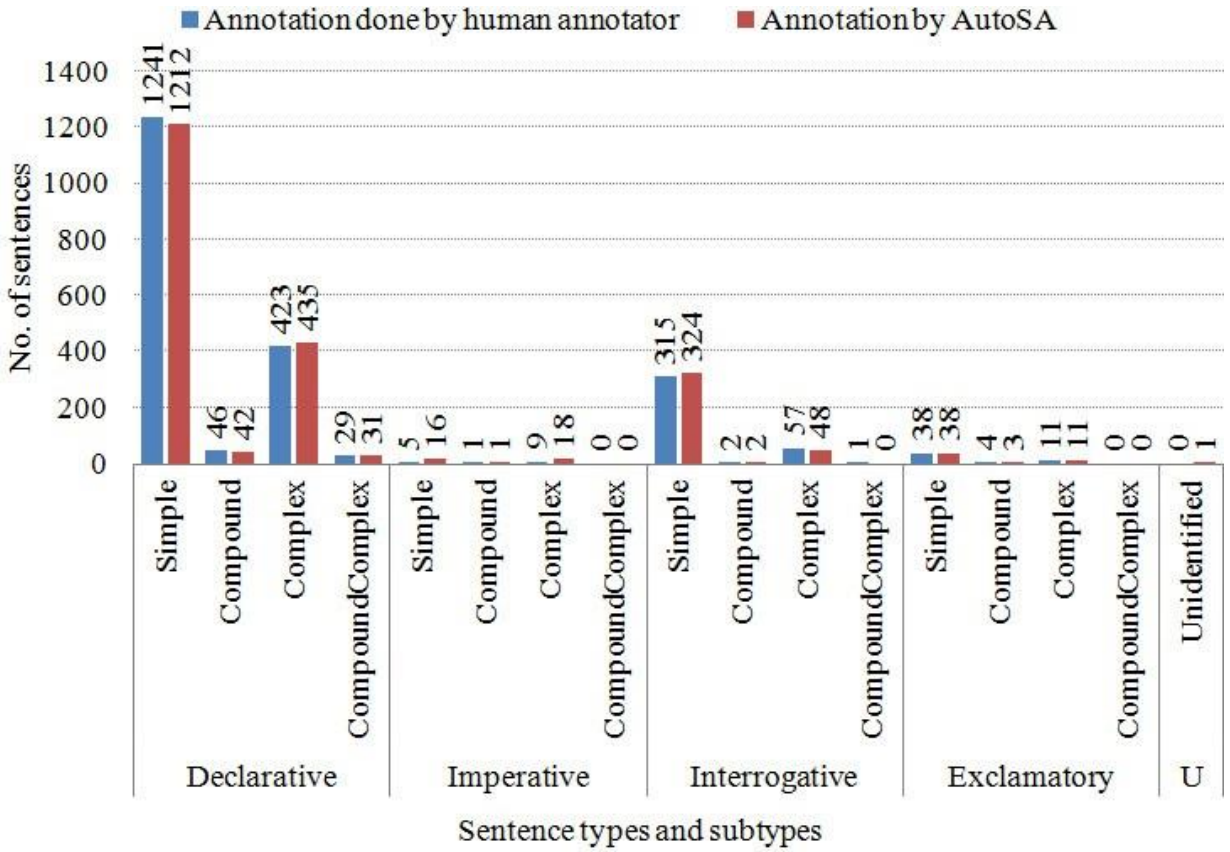


Figure 6: Manual Vs. Automated annotation of 2182 sentences from Tatoeba Project Corpus

3.1.3 Evaluation Results

There were differences between manual annotation and automatic annotation for 4% of sentences. When we closely analyzed these sentences, we found that for 3% of sentences there was error in manual annotation whereas in case of annotation done by AutoSA the error was 1%. Figure 6 shows the distribution of the annotations under each type and subtype of the 2182 sentences done by human annotator Vs. AutoSA; one can see that the annotation under each type and subtype of the sentences done by AutoSA are closer to those done by human annotator. Further, the results shown in Table 7 and Table 8, show that the false positives (FP) and false negatives (FN) were more for the annotation done by the human annotator than those done by AutoSA resulting in the higher precision and recall obtained by AutoSA as compared to the human annotator.

3.1.4 Validity Considerations

The threats to validity in our case study and the strategies used to deal with those threats [29,30] are as follow:

Internal validity: The manual annotations of the sentences with their types and subtypes were carefully done by the first author by thoroughly examining each sentence. The annotations were checked by the other two authors of the paper to evaluate the precision and recall of the manual annotations (i.e. the annotations done by the first author). The same sentences were then automatically annotated with their types and subtypes by the tool *AutoSA* and the annotations done by the tool were then compared with the manual annotations to evaluate the precision and recall of the approach. Hence, there was no comparison bias.

In a rule based approach, one of the threats to internal validity is overfitting. Since the validation set (the set of sentences which were used in the evaluation of the approach) was different from the training set (the set of sentences used for framing the rules to identify the types and subtypes of the sentences) and the number of correct classifications were very high in the validation, the threat to overfitting was minimized. The training set was composed of the sentences collected from sources such as English grammar books, research papers and websites. The testing set was composed of the sentences taken from Tatoeba project corpus.

External validity: The external validity of a study is related to the ability to generalize the results. One of

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 7: Results - Annotation of Tatoeba Corpus by human annotator

S#	Sentence Type	Sentence Subtype	Na	FP	FN	Nc	Nca	Recall	Precision
1	Declarative	Simple	1241	37	7	1211	1204	**99.4**	**97.0**
2	Declarative	Compound	46	5	–	41	41	**100.0**	**89.1**
3	Declarative	Complex	423	12	27	438	411	**93.8**	**97.2**
4	Declarative	CompCX	29	–	3	32	29	**90.6**	**100.0**
5	Imperative	Simple	5	–	10	15	5	**33.3**	**100.0**
6	Imperative	Compound	1	–	–	1	1	**100.0**	**100.0**
7	Imperative	Complex	9	–	7	16	9	**56.3**	**100.0**
8	Imperative	CompCX	0	–	–	0	0	–	–
9	Question	Simple	315	4	12	323	311	**96.3**	**98.7**
10	Question	Compound	2	–	–	2	2	**100.0**	**100.0**
11	Question	Complex	57	12	4	49	45	**91.8**	**78.9**
12	Question	CompCX	1	–	–	1	1	**100.0**	**100.0**
13	Exclamation	Simple	38	–	–	38	38	**100.0**	**100.0**
14	Exclamation	Compound	4	–	–	4	4	**100.0**	**100.0**
15	Exclamation	Complex	11	–	–	11	11	**100.0**	**100.0**
16	Exclamation	CompCX	0	–	–	0	0	–	–
17	Unidentified	Unidentified	0	–	–	0	0	–	–
Total Sentences=			**2182**			**Average=**		**90.1**	**97.2**

CompCX=Compound-Complex, Na= Number of annotated sentences
FP=False positive, FN=False negative, Nc= Number of sentences in the corpus =Na-FP+FN
Nca= Number of correctly annotated sentences, Recall= Nca/Nc, Precision= Nca/Na

Table 8: Results - Annotation of Tatoeba Corpus by AutoSA

S#	Sentence Type	Sentence Subtype	Na	FP	FN	Nc	Nca	Recall	Precision
1	Declarative	Simple	1212	2	1	1211	1210	**99.9**	**99.8**
2	Declarative	Compound	42	–	1	43	42	**97.7**	**100.0**
3	Declarative	Complex	435	2	4	437	433	**99.1**	**99.5**
4	Declarative	CompCX	31	–	1	32	31	**96.9**	**100.0**
5	Imperative	Simple	16	1	–	15	15	**100.0**	**93.8**
6	Imperative	Compound	1	–	–	1	1	**100.0**	**100.0**
7	Imperative	Complex	18	2	–	16	16	**100.0**	**88.9**
8	Imperative	CompCX	0	–	–	0	0	–	–
9	Interrogative	Simple	324	–	1	325	324	**99.7**	**100.0**
10	Interrogative	Compound	2	–	–	2	2	**100.0**	**100.0**
11	Interrogative	Complex	48	1	–	47	47	**100.0**	**97.9**
12	Interrogative	CompCX	0	–	–	0	0	–	–
13	Exclamatory	Simple	38	–	–	38	38	**100.0**	**100.0**
14	Exclamatory	Compound	3	–	–	3	3	**100.0**	**100.0**
15	Exclamatory	Complex	11	–	–	11	11	**100.0**	**100.0**
16	Exclamatory	CompCX	0	–	–	0	0	–	–
17	Unidentified	Unidentified	1	–	–	1	1	**100.0**	**100.0**
Total Sentences=			**2182**			**Average=**		**99.5**	**98.6**

CompCX=Compound-Complex, Na= Number of annotated sentences
FP=False positive, FN=False negative, Nc= Number of sentences in the corpus =Na-FP+FN
Nca= Number of correct annotated sentences, Recall= Nca/Nc, Precision= Nca/Na

The threats to external validity is the size of the study and the number of the representatives of the population. Though, the number of sentences used in the study are not very large. We tried to minimize the threat to external validity by randomly choosing 2182 sentences taken from Tatoeba Project Corpus.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

As the training set and validation set were different, the higher values of precisions and recalls obtained in the validation study imply that the completeness is also significantly high.

3.2 Case Study 2 : A part of Penn Treebank Corpus

The objective of the study was to evaluate the precision and recall of the approach in identifying the types (Declarative, Imperative, Interrogative or Exclamatory) and subtypes (Simple, Compound, Complex or Com-poundComplex) of the sentences collected from Penn Treebank Corpus.

Penn Treebank Corpus [27, 31] is a large corpus of sentences taken from Wall Street Journal, Brown Corpus, Switchboard and ATIS, which are annotated with POS-tags and parse trees. It is mainly used for training, testing and evaluation of different NL parsers. We were able to obtain the 5% sample (1650 sentences) of the Penn Treebank Corpus from[10].

3.2.1 Design of the study

As observed in Case Study 1, the number of false positives and false negatives in the annotations done by the human annotator were more than those done by AutoSA, so in this case study, instead of manually annotating the sentences and then comparing them with annotations done by AutoSA, we decided to get them annotated by AutoSA and then to check the annotations to evaluate the precision and recall of AutoSA.

The independent variables were the types and subtypes of the sentences identified by the tool, and the dependent variables were the precision (P) and recall (R) with which the types and subtypes were identified by the tool.

3.2.2 Execution of the study

The 1650 sentences of the Penn Treebank Corpus were given as input to *AutoSA*. The output (i.e. the sentences annotated with their types and subtypes) was exported by *AutoSA* in an Excel sheet. The time required in automatic annotation was less than 17 minutes (When AutoSA was executed on a computer with Intel Core i5 2.6 GHz CPU, 4GB RAM and 32 bit Windows 8 OS).

The annotations done by the tool were then carefully checked by the first author. The type I (false positive) and type II (false negatives) errors were recorded in the Excel sheet. The annotations and the errors were then cross checked by the other two authors of the paper. The results are shown in Table 9.

Table 9: Results - Annotation of Penn Treebank Corpus by AutoSA

S#	Sentence Type	Sentence Subtype	Na	FP	FN	Nc	Nca	Recall	Precision
1	Declarative	Simple	737	1	6	742	736	**99.2**	**99.9**
2	Declarative	Compound	45	2	–	43	43	**100.0**	**95.6**
3	Declarative	Complex	731	–	5	736	731	**99.3**	**100.0**
4	Declarative	CompCX	72	1	–	71	71	**100.0**	**98.6**
5	Imperative	Simple	11	4	–	7	7	**100.0**	**63.6**
6	Imperative	Compound	3	–	–	3	3	**100.0**	**100.0**
7	Imperative	Complex	6	2	–	4	4	**100.0**	**66.7**
8	Imperative	CompCX	2	–	–	2	2	**100.0**	**100.0**
9	Interrogative	Simple	2	–	–	2	2	**100.0**	**100.0**
10	Interrogative	Compound	0	–	–	0	0	–	–
11	Interrogative	Complex	2	–	–	2	2	**100.0**	**100.0**
12	Interrogative	CompCX	0	–	–	0	0	–	–
13	Exclamatory	Simple	0	–	–	0	0	–	–
14	Exclamatory	Compound	0	–	–	0	0	–	–
15	Exclamatory	Complex	0	–	–	0	0	–	–
16	Exclamatory	CompCX	0	–	–	0	0	–	–
17	Unidentified	Unidentified	39	3	2	38	36	**94.7**	**92.3**
Total Sentences=			**1650**		**Average=**			**99.4**	**92.4**

CompCX=Compound-Complex, Na= Number of annotated sentences
FP=False positive, FN=False negative, Nc= Number of sentences in the corpus =Na-FP+FN
Nca= Number of correct annotated sentences, Recall= Nca/Nc, Precision= Nca/Na

3.2.3 Validity Considerations

The threats to validity in our case study and the strategies used to deal with those threats [29, 30] are as follow:

[10]https://github.com/teropa/nlp/tree/master/resources/corpora/treebank (accessed March 27, 2016)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Internal validity:
The annotation of the sentences done by *AutoSA* were checked by the first author of the paper and then cross checked by the other two authors to evaluate the precision and recall. In this study also, the validation set is different from the training set. The training set was composed of the sentences collected from sources such as English grammar books, research papers and websites, and the validation set was composed of the sentences taken from a well known corpus in linguistic domain "Penn TreeBank Corpus". The number of correct classifications were very high also in this study. Hence, the threat to overfitting was minimized.

External validity: The external validity of a study is related to the ability to generalize the results. One of the threats to external validity is the size of the study and the number of the representative of the population. The testing set was an unknown corpus (corpus not used for training *AutoSA*) consisting of 1650 sentences taken from the Penn Treebank Corpus.
As the training set and validation set were different, the higher values of precisions and recalls obtained in the validation study imply that the completeness is also significantly high.

4 DISCUSSION

The training, testing and evaluation of the many NLP applications such as Information Extraction, Sentence Simplification, Text Summarization, Machine Translation and Automated Conversation Agent can better be done if corpora containing sentences annotated with the aforementioned types and subtypes are available. However, the available free/open corpora of English sentences are not annotated with such constructs. The corpora annotated with sentence types and subtypes can be constructed manually, but building large corpora using human annotators is a time consuming and error prone task [31]. Similar was our observation when we manually annotated 2400 English sentences of Tatoeba Corpus with the types and the subtypes of the sentences. Hence, the need of automated tools like *AutoSA* is highly desired.

To the best of our knowledge, the presented automated approach is the first one to classify the sentences into the types (Declarative, Imperative, Interrogative and Exclamatory) and subtypes (simple, compound, complex and compound-complex). The benefits of the presented approach are two folds: First, the approach can be used by NLP applications such as Information Extraction, Sentence Simplification, Text Summarization and Machine Translation to classify the input sentences into their types and subtypes so that they can process the sentences accordingly to correctly extract the relevant information from the sentences. Secondly, it can be used to build large corpus of English sentences annotated with their types and subtypes. The proposed approach along with the approaches for detecting grammatical errors [17–19] can be used to develop tool supports that the reviewers can use to analyse and evaluate various literature documents.

We present the following corpora of English sentences annotated with sentence types and subtypes: i) Tatoeba Project Corpus, ii) a part of the Penn Tree Bank Sentence Corpus. The annotated corpora can be used in training, testing and evaluation of many NLP applications such as those mentioned above. The tool support AutoSA can be used to build more such corpora. The details of the approach, the case study, and the annotated sentences of the above two corpora are made available at[11] for the interested readers. The interested users can also download the tool AutoSA from the same URL.
The limitations of the proposed approach are:

1) The approach uses a natural language parser to process the sentences in UCSs, hence the accuracy of the approach is inherently bounded by the accuracy of the NL parser.

2) It does not check grammatical errors in the sentences which may result in miss-classification of the sentences that have grammatical errors. Such error checking can easily be incorporated by including a pre-processing step for this operation.

3) The approach currently can not identify types of the sentences written in direct speech and those involving subject verb inversion.
Some sentences were misclassified by the approach because of the incorrect parse trees generated by the parser (the main cause of that was the incorrect POS-tags assigned by the parser to some words in those sentences). For instance, the sentence "*Friends of Education rates South Carolina one of the worst seven states in its study on academic cheating.*" of Penn Treebank Corpus was annotated as Unidentified. Its parse tree generated by the parser is shown in Figure 7; here, the word "rates" is assigned incorrect POS-tag "NNS";

[11]http://serg.iiitdmj.ac.in/tools/AutoSA (accessed March 27, 2016)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

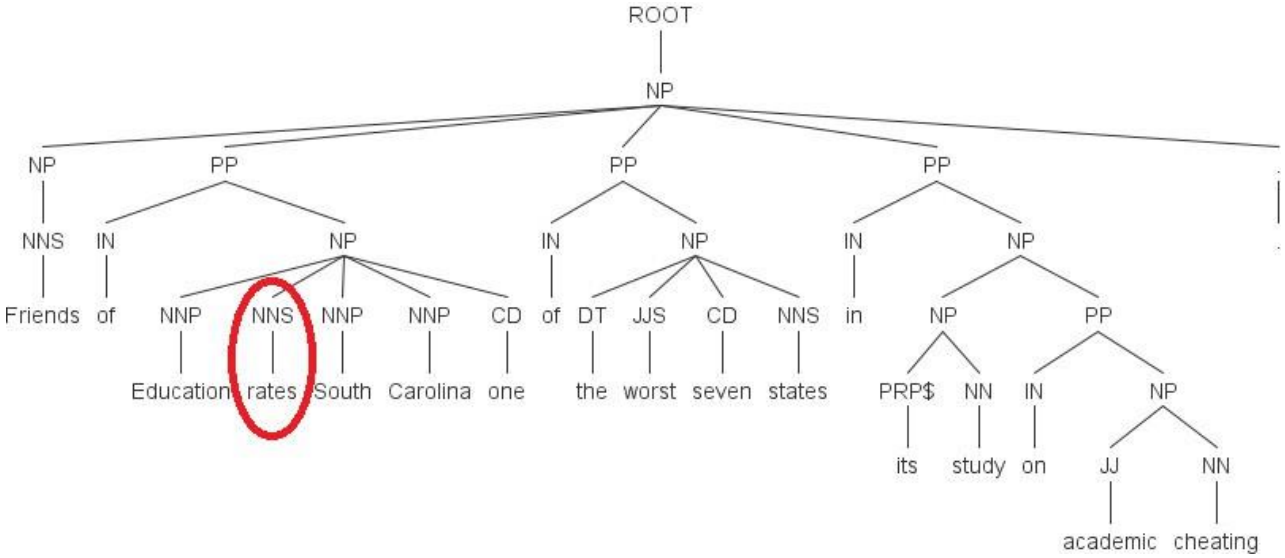


Figure 7: Incorrect parse tree generated by the parser for sentence "*Friends of Education rates South Carolina one of the worst seven states in its study on academic cheating.*"

its correct POS-tag in this context must be "VBZ". The other sentences which were annotated as Unidentified include sentence fragments (eg., *Not this year.*, *Like healthy regulatory capital.*), phrases (eg., *The next province.*), some incorrectly formed sentences which are common in newspapers but not in formal writing (eg., *So did the Federal Reserve Board 's industrial-production index.*, *The reason : the refusal of Congress to give federal judges a raise .*) and some incorrectly punctuated sentences (eg., *The student surrendered the notes , but not without a protest.*) The details of all the misclassified sentences of the Tatoeba Corpus and the Penn TreeBank Corpus are given in Appendix A and Appendix B, respectively.

5 TOOL AUTOSA (AUTOMATIC SENTENCE ANALYSER)

We prototyped the proposed approach in a GUI based tool *Automatic Sentence Analyser (AutoSA)* implemented in Java 1.6 using Eclipse Indigo IDE release 3.7.0. *AutoSA* uses the Stanford NLP parser APIs version 2.0.4[12] for parsing the sentences. It uses Apache POI API 3.9[13] for reading/writing the input/output Excel files. It provides a GUI interface to the user and provides a menu based option to read sentences from the inputs in many formats such as plain texts, Word documents and Excel sheets. It displays the output (i.e., the annotated sentences of each type and subtype) in a tabular form, and also exports the annotated sentences (corpus) in an Excel sheet.

6 RELATED WORK

The approach proposed by [14] applies unsupervised, statistical techniques: Similarity Approach, Naive Bayes classifier and multiple Naive Bayes classifiers on features such as words bigrams, trigrams, POS, polarity to classify sentences as fact, opinion or uncertain, and further to classify opinion into: positive, negative, neutral, mixed or uncertain. The approach of [16] uses Naive Bayes classifier to classify the sentences into two types: subjective (which represent opinions or emotions) and objective (which represent factual information). It uses subjective clues as feature for the classification.

[17] proposed an approach which uses a discriminative language model and an online learning algorithm proposed by [32] to classify the input sentences as correct or incorrect. [18] proposed an approach to classify the input sentences into two classes: erroneous and correct, using pattern discovery and supervised learning models. During training, the learning model is built by automatically extracting labeled sequential patterns (LSPs) from a sets of the erroneous and the correct sentences. The LSPs are used as features along with some other linguistic features such as lexical collocation, language model, syntactic score and function density to the classify the sentences as erroneous or correct. The approach is able to detect errors such as grammar errors, sentence structure errors and lexical errors. [19] described a method to classify the sentences in English into two types: well formed and ill formed. The method implies machine learning that uses part-of-speech n-gram frequencies, the grammatical judgments provided by a hand-crafted generative grammar of English and the output of probabilistic parsers of English.

[12]http://nlp.stanford.edu/software/lex-parser.shtml (accessed March 27, 2016)
[13]http://poi.apache.org/ (accessed March 27, 2016)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

[20] presented a semi-supervised approach that uses features such as phrases and dependency trees method proposed by [33] to identify speech acts from email and forum documents. It classify the sentences as: Accept response, Acknowledge and appreciate, Action motivator, Polite mechanism, Rhetorical question, Open-ended question, Or-clause question, Wh-question, Yes-no question, Reject response, Statement or Uncertain response. The approach proposed by [21] uses a binary SVM classifier to detect the presence or absence of requests in email messages using the features like message length, number and percentage of capitalised words, number of non alpha-numeric characters, whether the subject line contains markers of email replies or forwards (e.g. Re:, Fw:), the presence of sender or recipient names, the presence of sentences that begin with a modal verb (e.g., might, may, should, would), the presence of sentences that begin with a question word (e.g, who, what, where, when, why, which, how), whether the message contains any sentences that end with a question mark, binary word unigram and word bigram features for n-grams that occur at least three times across the training set.

The approach of [22] detects question-answer pairs from email conversations. It detects questions using features such as POS tags for the first five terms, POS tags for the last five terms, length of utterance and POS-bigrams. It can only detect interrogative questions. [23] proposed an approach for detecting question-answer pairs from online forums sentences. It obtains labeled sequential patterns from POS-tags of the sentences to detect questions in a forum thread, and uses a graph based propagation method to find answers for the questions in the same thread. Their intent is to use the extracted question-answer pairs to enhance the knowledge base of Question-Answering services. The approach proposed by [24] detects question sentences from Community-based Question Answer services (CQA). Certain questions can not be extracted using only lexical feature (sequential POS patterns) so it uses syntactic feature (parse tree patterns) along with the lexical features for extraction. The approach of [25] detects questions in Twitter micro-text [14]. When writing tweets, people don't care of correctness but use short and creative text that are highly influenced by the topic, culture and other attributes. Therefore the approach uses a different grammar consisting of 500 rewrite rules for identifying questions from such sentences. [26] proposed an approach for detecting question-answer pairs in email conversations that include imperative and declarative questions. The approach uses three algorithms for question detection in an email: Naive, [22] and Regex.

[13] proposed a method to classify the sentences into three classes: Declarative, Interrogative and Imperative. The intent is that the identified sentence class of a spoken phrase would help an automated conversation agent to provide a proper response. Their system used three binary support vector machines for each sentence class and used POS ngrams, dependency patterns, constituency parsing and word clustering to train the classifier. From the dataset of 1050 sentences (400 short and simple sentences constructed manually by one annotator which were noise free and 650 possibly noisy sentences automatically collected from sites "uselessfacts.net" and "wikihow.com, and other internet sources such as news, articles, the enron email dataset, reviews, transcriptions and forums discussions), 1044 sentences were used for training and 232 sentences were used for testing the classifier. Their presented results show that their approach achieved maximum precisions and recalls with the feature (POS+WRD+Clus) specifically, the precisions and recalls of .75 and .83 for declarative, .88 and .70 for imperative, and .86 and .92 for interrogative sentences, respectively.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed an automated approach to classify the English sentences into their types (Declarative, Imperative, Interrogative and Exclamatory) and subtypes (Simple, Compound, Complex and Compound-Complex). We prototyped the approach in a GUI based tool *AutoSA*. The results of the two case studies conducted for validation of the approach showed that AutoSA annotated the 2182 sentences of the Tatoeba Corpus with the precision of 98.6% and the recall of 99.5%, and annotated the 1650 sentences of the Penn Treebank Corpus with the precision of 92.4% and the recall of 99.4%.

We present the following corpora of English sentences annotated with sentence types and subtypes: i) Tatoeba Project Corpus, ii) a part of the Penn Tree Bank Sentence Corpus. The annotated corpora can be used in training, testing and evaluation of many NLP applications such as Information Retrieval, Sentence Simplification, Language Translation and Text Summarization. Moreover, the tool *AutoSA* can be used to build more such corpora annotated with the types and subtypes of the sentences from any text documents.

The proposed approach can further be extended for some other classifications of sentences: i) Direct speech vs Indirect speech sentences, ii) Active voice vs Passive voice sentences.

[14]Very short text, typically of 140-character length

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 11, November 2021
DOI: 10.17148/IJARCCE.2021.101132

REFERENCES

[1] J. Eastwood, *Oxford guide to English grammar*. Oxford University Press, 1994.
[2] S. Greenbaum, *The Oxford English Grammar*, vol. 652. Oxford University Press Oxford, 1996.
[3] M. Verspoor and K. Sauter, *English sentence analysis: An introductory course*. John Benjamins Publishing, 2000.
[4] B. Aarts, *English syntax and argumentation*. Palgrave Macmillan, 2013.
[5] J. S. Thakur and A. Gupta, "Anmodeler: a tool for generating domain models from textual specifica-tions," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pp. 828–833, ACM, 2016.
[6] J. S. Thakur and A. Gupta, "Automatic generation of analysis class diagrams from textual specifications," tech. rep., Software Engineering Research Group, IIITDM Jabalpur, India, February 2016.
[7] J. S. Thakur and A. Gupta, "Identifying domain elements from textual specifications," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pp. 566–577, ACM, 2016.
[8] J. S. Thakur and A. Gupta, "Automatic generation of sequence diagram from use case specification," in *Proceedings of the 7th India Software Engineering Conference*, p. 20, ACM, 2014.
[9] T. Yue, L. C. Briand, and Y. Labiche, "atoucan: An automated framework to derive UML analysis models from use case models," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 24, no. 3, p. 13, 2015.
[10] J. S. Thakur, *Identifying Domain Models from Textual Specifications*. PhD thesis, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, INDIA, 2017.
[11] R. M. Iyer and M. Ostendorf, "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models," *IEEE Transactions on speech and audio processing*, vol. 7, no. 1, pp. 30–39, 1999.
[12] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4,pp. 403–434, 2001.
[13] K. Quinn and O. Zaiane, "Identifying questions & requests in conversation," in *Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering*, p. 10, ACM, 2014.
[14] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, Association for Computational Linguistics, 2003.
[15] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 105–112, Association for Computational Linguistics, 2003.
[16] J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts," in *Computational Linguistics and Intelligent Text Processing*, pp. 486–497, Springer, 2005.
[17] D. O. J. Tsujii Jun'ichi, "A discriminative language model with pseudo-negative samples," *ACL 2007*,p. 73, 2007.
[18] G. Sun, X. Liu, G. Cong, M. Zhou, Z. Xiong, J. Lee, and C.-Y. Lin, "Detecting erroneous sentences using automatically mined sequential patterns," in *Annual Meeting-Association for Computational Linguistics*, vol. 45, p. 81, 2007.
[19] J. Wagner, J. Foster, and J. van Genabith, "Judging grammaticality: Experiments in sentence classifica-tion," *CALICO Journal*, vol. 26, no. 3, pp. 474–490, 2013.
[20] M. Jeong, C.-Y. Lin, and G. G. Lee, "Semi-supervised speech act recognition in emails and forums," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1250–1259, Association for Computational Linguistics, 2009.
[21] A. Lampert, R. Dale, and C. Paris, "Detecting emails containing requests for action," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 984–992, Association for Computational Linguistics, 2010.
[22] L. Shrestha and K. McKeown, "Detection of question-answer pairs in email conversations," in *Proceedings of the 20th international conference on Computational Linguistics*, p. 889, Association for Computational Linguistics, 2004.
[23] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun, "Finding question-answer pairs from online forums," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 467–474, ACM, 2008.
[24] K. Wang and T.-S. Chua, "Exploiting salient patterns for question detection and question retrieval in community-based question answering," in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1155–1163, Association for Computational Linguistics, 2010.
[25] K. D. Dent and S. A. Paul, "Through the twitter glass: Detecting questions in micro-text.," in *Analyzing Microtext*, 2011.
[26] H. Kwong and N. Yorke-Smith, "Detection of imperative and declarative question–answer pairs in email conversations," *AI Communications*, vol. 25, no. 4, pp. 271–283, 2012.
[27] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: an overview," in *Treebanks*, pp. 5–22, Springer, 2003.
[28] B. Santorini and M. A. Marcinkiewicz, "Bracketing guidelines for the penn treebank project," *Unpublished manuscript, Department of Computer and Information Science, University of Pennsylvania*, 1991.
[29] C. Wohlin, M. Höst, and K. Henningsson, "Empirical research methods in software engineering," in *Empirical Methods and Studies in Software Engineering*, pp. 7–23, Springer, 2003.
[30] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in software engineering*. Springer Publishing Company, Incorporated, 2012.

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 11, November 2021
DOI: 10.17148/IJARCCE.2021.101132

[31] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
[32] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algo-rithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
[33] T. Kudo and Y. Matsumoto, "A boosting algorithm for classification of semi-structured text.," in *EMNLP*, vol. 4, pp. 301–308, 2004.

APPENDIX

A Misclassified Sentences Tatoeba Corpus

Table 10 shows the sentences (out of 2182 sentences used in the validation study) of the Tatoeba Corpus which were misclassified by AutoSA along with the the possible reasons for the misclassification.

Table 10: Misidentified sentences Tatoeba Corpus

S#	Sentence	Expected Type and Subtype	Identified Type and Subtype	Reason for misidentification
1	Haven't you got any money?	Interrogative simple	Interrogative Complex	Incorrect parse tree generated by the parser.
2	What you have said doesn't apply to you.	Declarative Complex	Imperative Complex	Needs one more pattern to identify declarative sentences which starts with a Wh-word.
3	What you don't have is better than what you do have.	Declarative Complex	Imperative Complex	Needs one more pattern to identify declarative sen-tences which starts with a Wh-word.
4	Ah! is an interjection.	Declarative	Imperative	Incorrect parse tree generated by the parser.
5	Poor is not the one who has too little, but the one who wants too much.	Simple Declarative Compound Complex	Simple Declarative Complex	The sentence is correctly identified by the tool as Declarative Complex. Here the sentence was incorrectly annotated by the human annotator as Declara-tive Compound-Complex
6	Our opinion is an idea which we have; our conviction an idea which has us.	Declarative Compound Complex	Declarative Complex	This sentence is grammatically erroneous as the independent clause after the semicolon doesn't contain verb phrase (probably "is") after the noun phrase (our conviction).
7	You may go anywhere you like.	Declarative Complex	Declarative Simple	Incorrect parse tree generated by the parser.
8	Wherever you go, you'll be welcomed.	Declarative Complex	Declarative Simple	Here the word "wherever" is incorrectly identified as verb in place of adverb by the parser. As a result, incorrect parse tree of the sentence is generated.

B Misclassified Sentences Penn Treebank Corpus

Table 11 shows the sentences (out of 1650 sentences used in the validation study) of the Penn Treebank Corpus which were misclassified by AutoSA along with the the possible reasons for the misclassification.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101132

Table 11: Misidentified sentences Penn Treebank Corpus

S#	Sentence	Expected Type and Subtype	Identified Type and Subtype	Reason for misidentification
1	He also said that after the charges , and " as- suming no dramatic fluctuation in interest rates , the company expects to achieve near- record earnings in 1990 . "	Declarative Complex	Declarative Simple	Ill formed sentence (puntuation errors)
2	The competition has grown more intense as big- ger banks such as Norwest Corp. of Minneapolis and Chemical Banking Corp. of New York extend their market-share battles into small towns across the nation .	Declarative Complex	Declarative Compound	Incorrect parse tree generated by the parser
3	" To avoid these costs , and a possible default , immediate action is imperative . "	Declarative Simple	Declarative Compound	Incorrect parse tree generated by the parser
4	The idea , of course : to prove to 125 corporate decision makers that the buckle on the Rust Belt is n't so rusty after all , that it 's a good place for a company to expand .	Declarative Complex	Declarative Compound- Complex	Incorrect parse tree generated by the parser
5	Partly to help clear the myriad obstacles fac- ing any overseas company trying to penetrate Japan , tiny Candela turned to Mitsui & Co. , one of Japan 's largest trading companies .	Declarative Simple	Imperative Simple	Incorrect parse tree generated by the parser
6	Performing loans .	Unidentified	Imperative Simple	Ill formed sentence
7	asks a female voice .	Unidentified	Imperative Simple	Ill formed sentence
8	Among them are differences in savings and investment rates , corporate structures and management , and government spending .	Declarative Simple	Imperative Simple	Ill formed sentence
9	What she did was like taking the law into your own hands . "	Declarative Complex	Imperative Complex	Needs one more pattern to iden tify declarative sentences which starts with a Wh-word.
10	Where they disagree is on the subject of U.S. direct investment in Japan .	Declarative Complex	Imperative Complex	Needs one more pattern to iden- tify declarative sentences which starts with a Wh-word.
11	Here are the Commerce Department 's latest figures for manufacturers in billions of dollars , seasonally adjusted .	Declarative Simple	Unidentified	The approach currently can not identify the types of Subject verb Inverted sentences
12	Friends of Education rates South Carolina one of the worst seven states in its study on	Declarative Simple	Unidentified	The parser incorrectly tagged "rates" as NNS in
13	academic cheating . The low-ball bids touch on issues central to the increasingly tense trade debate .	Declarative Simple	Unidentified	place of VBZ The parser incorrectly tagged "touch" as NN in place of VB

