

ARTICLE

Pattern-Based Syntactic Simplification of Compound and Complex Sentences

Archana Praveen Kumar, Ashalatha Nayak, Manjula Shenoy K, Roshan Jacob Manoj,
and Akansha Priyadarshi

Manipal Institute of Technology,
Manipal Academy of Higher Education
archana.kumar@manipal.edu, asha.nayak@manipal.edu, manju.shenoy@manipal.edu, roshan.jacob@learner.manipal.edu,
akansha.priyadarshi@learner.manipal.edu

(Received xx xxx xxx; revised xx xxx xxx; accepted xx xxx xxx)

Abstract

In the field of Natural language processing many of the tasks like text summarization, anaphora resolution, machine translation are the most preliminary steps done for a given application like multiple-choice question generation, sentiment analysis and many more. These tasks are done automatically by available tools. Most of these tools are trained artificially with simple sentences and therefore they work with high accuracy for such sentences. When the dataset used for these kind of tasks comprise of either complex or compound sentences the performance of automatic tools lack accuracy. So there is a need to make use of a preprocessing step which convert both complex and compound sentences into simple sentences. A system dedicated for simplifying the sentences from both compound and complex sentences has not been developed so far. Our proposed system initiates the task of simplifying the sentences by identifying its type, applying the pattern and then converting them into simple sentences. The experiment demonstrates that the proposed system yields promising results which can aid the automatic tools.

Keywords: Syntactic Simplification, Simple Sentences, Compound Sentences, Complex sentences, Pattern-Based, Context Free Grammar.

1. Introduction

Tasks like anaphora resolution, machine translation, text summarization are some of the most preliminary steps done for Natural Language Processing (NLP) applications like question-answering, multiple-choice question generation, sentiment analysis, opinion mining, etc. The results of these preliminary tasks need to be accurate and precise for a given natural language. Manual efforts in handling these tasks are very efficient but consumes time and are tedious. In order to reduce these factors research has contributed and provided with many online automatic tools for each of these tasks (R. Sukthanker, S.Poria, E. Cambria, and R. Thirunavukarasu (2018)) and (A. M. Azmi and N. I. Altmami (2018)).

The tools either make use of machine learning (V. Nq (2015)) approaches or statistical approach or deep learning techniques (R. Sukthanker, S.Poria, E. Cambria, and R. Thirunavukarasu (2018)), (S. S. Yadav and S. Bojewar (2015)) to complete the tasks. Though these tools perform accurately for simple sentences, they lack performance when the data encountered comprises of compound or complex sentences. The reason is that the compound or complex sentences comprise

of dependent clauses, co-reference embedded within them due to which the performance of the automatic tools reduce (S. Wubben, A. V. D. Bosch, and E. Krahmer (2012)), (C. Poornima, V. Dhanalakshmi, K. M. Anand and K. P. Soman (2011)) and (D. Vickrey and D. Koller (2008)). To achieve accurate results there is a need of an input corpus for the tools comprising only simple sentences (A. Bawakid and M. Oussalah (2011)), (F. A. Tarouti, J. Kalati, and C. McGrory (2015)) and (M. Majumder and S. K. Saha (2015)). Therefore there is a necessity of extracting simple sentences from compound or complex sentences in order to use these tools for the required NLP tasks.

A short and uncomplicated sentence composed of a group of words conveying a complete sense is known as a simple sentence (T. B. McArthur and F. McArthur (1999)) and (B. Backman (2003)). Simple sentences comprise of one independent clause having only one subject, verb, and object (G. Lutz and D. Stevenson (2005)) which conveys meaning to the reader or listener (F. Obrecht (1999)). E.g. "Mary has a little lamb". If the sentence does not represent a complete thought then it might comprise of one or more dependent clauses that require the support of an independent clause in the sentence fragment (T. P. Klammer, R. M. Shultz, and A. D. Volpe (2007)). E.g. "Mary has a little lamb, whose fleece was white as snow". In the example above dependent clause -'whose fleece was white as snow' cannot express the meaning without the independent clause -'Mary has a little lamb'. Such sentences which have one or more dependent clauses along with the independent clause are complex (M. Majumder and S. K. Saha (2015)). Sentences which have two independent clauses and are connected by coordinating conjunctions are compound sentences (J. Frost (2020)).

for and yet ka problems
not mentioned

Compound sentences are connected by coordinating conjunction like - 'for', 'and', 'nor', 'but', 'or', 'yet', 'so'. The conjunctions together represent 'FANBOYS' stating for the connectors between two or more independent clauses (J. Sevastopoulos (2019)). E.g. "The boys sang and the girls danced". Compound sentences if not connected by the conjunction can also be separated by a semicolon or full stop (J. Frost (2020)). Complex sentences are sentences that have one independent clause and at least one dependent clause. Whereas a dependent clause has a subject along with a verb and cannot express a complete thought. Both the independent and dependent clauses are joined in any order with subordinating conjunction like 'after', 'while', 'since' and so on. If the dependent clause comes before the independent clause they are usually separated by a comma (<https://examples.yourdictionary.com/> (2020)). Compound-Complex sentences are the ones having at least two independent clauses and one or more dependent clauses (B. Das, M. Majumdar and S. Phadikar (2018)). The compound and complex sentences have complexities that need to be addressed to convert them into simple sentences. Therefore the literature review suggests that sentence simplification can alleviate the problems of the tools when they deal with such sentence complexities (R. Chandrasekar, C. Doran, and B. Srinivas (1996)), (S. Jonnalagadda and G. Gonzalez (2010)), (Siddharthan (2003)), (Y. Peng, C. O. Tudor, M. Torii, C. H. Wu, and K. Vijay-Shanker (2012)) and (D. R. Ch and S. K. Saha (2018)).

Text simplification or sentence simplification has been done either by lexical simplification or by syntactic simplification. Lexical Simplification is done by identifying and replacing difficult phrases or words while syntactic simplification deals with converting the complex structure of the sentence to simple syntactic structures (C. Scarton, G. Paetzold, and L. Speci (2018)). Despite of having rich research history in the NLP applications, sentence simplification has been explored in very few research papers. Therefore we find simplification approaches confined to using a set of rules or checking the occurrence of pre-defined features for a given domain thus making them domain reliant (M. Heilman and N. A. Smith (2010)). Some of the approaches have been applied to a particular language (L. Brouwers, D. Bernhard, A. L. Ligozat, and T. Francois (2014)), and (S. K. D. Nikita and S. K. Sharma (2015)). Major of the techniques have been applied to either compound (S. R. Savanur and Dr. R. Sumati (2018)), and (Z. Zhu, D. Bernhard, and I. Gurevych

(2010)) or complex sentences (L. Brouwers, D. Bernhard, A. L. Ligozat, and T. Francois (2014)). Although the previous works were not in vain, there are certain shortcomings concerning with the grammatical correctness of the sentence after simplification which needs to be tackled.

The motivation of this research is to make a generic system that can syntactically simplify both compound and complex sentences exploiting the grammar patterns. The grammar rules govern the occurrence of words in the sentences which can be used for simplifying the structure of the sentences. These grammar rules help in restructuring the sentences thus preserving the grammatical correctness. In this paper, we propose an approach that exploits patterns of the original sentence to identify the type of sentence, split and extract the simple sentences. The proposed method can:

- Identify compound, complex and simple sentences based on patterns of grammar
- Analyse and correct the sentence structure based on the type of sentence
- Split and rephrase the sentence grammatically using the original words to convey the meaning of the original sentence

The rest of the article is subdivided as follows. The related works are presented in Section 2. Section 3 gives a brief explanation of the background knowledge required to solve the problem. The methodology is explained in Section 4. Section 5 provides the analysis and discussions of the results. Finally, we conclude our work in Section 6.

2. Related Work

Sentence simplification, a preliminary task required for NLP tasks has been often discussed from a linguistic point of view. A possible requirement of simplification is to get an easy and simple version of a given sentence. Based on this, existing approaches have explored parsing and dependency linkages of a given language to syntactically simplify sentences. The simplification has been also exploited techniques of machine learning which include statistical machine translation and Conditional Random Field (CRF) to decompose sentences. Some approaches made use of semantic role labeling and others have focussed on the extraction of relations between the entities. In comparison to the related work, our approach exploits the patterns of grammar for each sentence which has paved a better accuracy **concerning the grammatical correctness of the decomposed sentence.**

The initial work on the syntactic simplification of a sentence was done by Chandrasekar et al. (R. Chandrasekar, C. Doran, and B. Srinivas (1996)) wherein the researcher has done two alternatives for full parsing to simplify the sentences. One approach groups noun and verb groups through Finite State Grammar while another approach generates dependency linkages by using the Super-tagging model. The research discusses that the two input representations done by the approach are useful for simplifying the sentence. This approach was done to compare which among the two methods was better for simplifying. Another research by Vickrey and Koller (D. Vickrey and D. Koller (2008)) have used semantic role labeling for simplifying the sentence. The sentence is simplified by keeping all arguments of the verb and removing the information outside the arguments and verbs. The approach failed to identify the direct object for some sentences. Zhu et al. (Z. Zhu, D. Bernhard, and I. Gurevych (2010)) have proposed a method wherein a Tree model is done to derive a parse tree for complex sentences. This model generates the parse tree by splitting, reordering, dropping and substitution operations. Then the sentence simplification is done by this model based on the statistical machine translation. This approach does not simplify compound sentences.

Another approach done by Heilman and Smith (M. Heilman and N. A. Smith (2010)) was based on rules called Simplified Factual Statement Extractor. This can extract multiple simple sentences from the text thus helping the generation of questions automatically. This experiment ignores the compound and complex kind of declarative sentences. Miwa et al. (M. Miwa, R. Satre, Y. Miyao and J. I. Tsujii (2010)) proposed the Sentence simplification technique focused on entities used for extracting relations. Their technique considers that the truth value of the relation is quite important than the sense of the target sentence. Therefore two rules are defined for the clause selection and entity phrase. The clause selection removes the noise before and after the relevant clause while the entity phrase rule simplifies the entity without changing the truth value of the relation. The limitation of this approach is some rules tend to change the modality of the sentence during simplification.

Syntactic simplification for French texts has been explored by Brouwers et al. (L. Brouwers, D. Bernhard, A. L. Ligozat, and T. Francois (2014)). This method uses two stages where in the first step all the possible simplifications for a given sentence are generated. The second step selects the best- simplified sentence by a rank satisfying certain criteria. The limitation is the pre-requisite of hand-crafted rules extracted manually from a French corpus. The system is confined to work only with predefined rules thus making it domain-dependent. Tarouti et al. (F. A. Tarouti, J. Kalati, and C. McGrory (2015)) proposed an approach using a method by name Simplified Statement Extraction (SSE) which decomposed the original sentence into small simplified sentences. The experiment yields simple sentences but with lower grammatical accuracy. Nikita and Sharma (S. K. D. Nikita and S. K. Sharma (2015)) identified complex sentences from Punjabi texts using CRF. All the accessible features that included interactions between sentences were noted by the framework using CRF. The approach is limited to the Punjabi language. Bidyut et al. (B. Das, M. Majumdar and S. Phadikar (2018)) have targeted simplification of complex-compound sentences by using the Stanford dependency parser to generate Modified Stanford Dependency (MSD) Structure derived from Basic Stanford Dependency (BSD) and Collapsed Stanford Dependency (CSD) structure of the parser. This method checks the number of subjects to decide whether the sentence is a simple sentence or not. Then using MSD removes the other set of dependencies for the sentences identified as not simple. Identifying the subject clause, traversing all the paths which link the subject clause the approach finds the linked words. After rearranging the words with the subject clause the simple sentences are extracted. The system has a limitation of generating incomplete sentences for some test cases of complex-compound sentences.

The grammatical structures for the different sentences have been explored by Umesh et al. (U. C. Jaiswal, R. Kumar, and S. Chandra (2009)). The paper gives analysis and structural representations for simple and compound sentences which has paved the way to explore patterns in the sentences in our research. The criteria for distinguishing simple from the compound and complex sentences have been explored by Klimova in a paper (B. F. Klimova (2013)). The paper has analyzed abstracts of papers to discover which type of sentences and clauses have been typically a part of the scientific abstracts. For this analysis, a syntactic-semantic classification has been carried out. This approach provides a way of distinguishing the simple from the compound and complex sentences with the number of verbs in the sentence. Findings are limited for sentences found in abstract parts of scientific texts. Sentence classification has paved an important improvement for sentiment classification which is contributed by Chen et al. (T. Chen, R. Xu, Y. He and X. Wang (2017)). The sentence classification for this kind of task is done by the divide and conquer approach followed by the sentiment classification of one-dimensional convolution neural network. Experimental results prove that sentence classification done before sentiment analysis has improved the analysis over several benchmarking datasets. This experiment is limited to sentences having opinions or sentiments. Different types of compound sentences and their

structures have been discussed by Sandhya and Sumati (S. R. Savanur and Dr. R. Sumati (2018)). This paper discusses the different conjunctions, correlative conjunctions and subordinating conjunctions in the sentences along with their order to illustrate the various types of declarative compound sentences in English. This paper paves a way to understand which all sentences can be considered as compound sentences in a given dataset used in the system. This approach is limited to compound sentences having sentiments or opinions.

Lexical simplification of the sentences has been tackled by Narayan et al. (S. Narayan, C. Gardent, S. B. Cohen, and A. Shimorina (2017)) using a technique for simplifying complex sentences called Split and Rephrase. The experiment is conducted with a dataset comprising of RDF(Resource Description Framework) triples using the Sequence-to-Sequence approach. RDF is the framework in which the data is represented as triples in the form of <subject><predicate><object> used for Semantic web. The limitation of the experiment is the sentence splitting points which are determined by probabilistic model predicts lower quality simplifications for a given sentence. Another approach is done by Coster and Kauchak (W. Coster and D. Kauchak (2011)) where for a given input sentence a simplified sentence is generated by utilizing a corpus. A new translation model which extends phrase-based machine translation approach is introduced to delete the phrases of the input text converting it to simple. The limitation in this approach is for some simplified sentences the grammar is incorrect.

Belder and Moens (J. D. Belder and M. F. Moens (2010)) have proposed an approach of text simplification for children wherein the difficult words are replaced by easier synonyms. This is done by sentence splitting and lexical simplification. The lexical simplification causes some problems which introduce certain errors while simplification. Narayan and Gardent (S. Narayan, C. Gardent, S. B. Cohen and A. Shimorina (2016)) have described another approach that has simplified the sentences without the use of corpus and rules. They have made an unsupervised approach that is competitive and effective in handling sentence simplification than their previous approaches. The limitation of the system is that it cannot simplify sentences that have parsing errors. Biran et al. (O. Biran, S. Brody, and N. Elhadad (2011)) devised a lexical model for sentence simplification which used two stages. The first stage denotes the extraction of rules comprising of ordered pairs of words (original, simplified) from the corpora. This stage also provides the similarity score between the words in the ordered pair. Based on the contextual information the second stage decides whether the rule needs to be applied. As the similarity score is between the original words and words occurring in the domain, the system works effectively only for words which occur frequently in the domain.

The literature discusses both syntactic and lexical approach but lacks a dedicated approach to simplify and split the compound as well as the complex sentences. Most of the earlier works concentrate on features of the dataset used, therefore, confine to the given domain. The main advantage of our approach lies in the fact that the grammar rules are exploited and hence the system can work for any domain. The previous approaches either target compound or complex sentences whereas our approach works for both compound and complex sentences thereby making it generic. In addition, our approach is the only approach that exploits grammatical rules of the given language it can, therefore, rephrase the sentences accurately in comparison to the previous works.

3. Background



There are different kinds of sentences i.e. **declarative**, **imperative** and **interrogative**. Each of the sentences follow a pattern of words that need to be grouped to form a complete thought. In linguistic terminology, every structure of the sentence in English and other natural languages can be represented in a mathematical form. This mathematical formalism in English is called the Context-Free-Grammar (CFG) or Phrase-Structure Grammar and is equivalent to what is termed as Backus Normal Form (BNF) (D. Jurafsky and H. James (1996)).

As per CFG, the structure of each sentence is encoded with a rule based on the pattern followed. CFG comprises of a set of rules or productions which help to express the ways how the set of symbols or words of the language can be ordered forming a pattern. For example to represent that an NP (noun phrase) can be composed of proper noun or determiner followed by a Nominal (can be one or more nouns) is easily represented in CFG as shown in 1a, 1b, and 1c.

$$NP \rightarrow Det\ Nominal \quad (1a)$$

$$NP \rightarrow ProperNoun \quad (1b)$$

$$Nominal \rightarrow Noun \mid Noun\ Nominal \quad (1c)$$

CFG can be used for two reasons: one for generating sentences and other for assigning structure to a sentence. So starting from the nonterminal on the left-hand side of the rule, the nonterminal symbols on the right-hand side of the rule derive the terminals or words of the sentence. Thereby deriving the parse tree for a given sentence.

Declarative sentences are sentences which follow the structure of subject-NP followed by a VP (verb phrase). Imperative sentences are sentences that follow the structure of the verb phrase in terms of command or request. Interrogative sentences are those categorized as in yes-no or Wh-kind of questions. Yes-no sentences begin with an auxiliary verb followed by a noun phrase and then a verb phrase. Wh-questions are similar to declarative sentences except for having Wh-kind noun phrase followed by the verb phrase.

CFG can represent sentence-level grammatical constructions in English for declarative as shown in 2a, imperative in 2b, yes-no question in 2c and wh-question type of sentences as in 2d. Here the nonterminal is S which stands for the sentence and each of the sentences can be derived using the production rules. Aux is the auxiliary verb while Wh-NP refers to noun subject containing wh-word e.g. what, who and so on.

$$S \rightarrow NPVP \quad (2a)$$

$$S \rightarrow VP \quad (2b)$$

$$S \rightarrow Aux\ NPVP \quad (2c)$$

$$S \rightarrow Wh - NPVP \quad (2d)$$

Each of the above symbols in the equations can be further divided into fine granular elements as given below.

$$S \rightarrow < subject > + < verb > \quad (3a)$$

$$< subject > \rightarrow < Noun/Pronoun > \text{ or} \quad (3b)$$

$$< subject > \rightarrow < Restrictor > < Noun/Pronoun > \text{ or} \quad (3c)$$

$$< subject > \rightarrow < Determiner > < Noun/Pronoun > \quad (3d)$$

$$< verb > \rightarrow < verb > \text{ or} \quad (3e)$$

$$< verb > \rightarrow < verb > < complement > \text{ or} \quad (3f)$$

$$< verb > \rightarrow < verb > < object > \quad (3g)$$

The scope of this research is targeted at declarative sentences. The declarative sentences can be categorized as simple, compound and complex sentences.

3.1 Compound Sentences

As compound sentences are bound to comprise of at least two independent clauses connected by conjunction, there is a rule which can be seen commonly in all the compound sentences. The rule helps to categorize the compound sentences into three types:

- (1) When each of the independent clauses comprises of one subject each performing different actions connected by conjunction e.g. 'Cats like to drink milk and dogs like to chew bone'
- (2) When the subject of an independent clause performs two actions connected by a conjunction e.g. 'Cats like to drink and play with the ball'
- (3) When there exists an independent clause connected with conjunction with another object that further extends the first independent clause e.g. 'Cats like to play with dogs and other cats'

CFG for compound sentences can be represented as follows:

$$S = < subject > + < verb > + < object > \quad (4a)$$

$$C = < subject1 > + < verb1 > + < object1 > + < CC > + < subject2 > + < verb2 > + < object2 > \quad (4b)$$

$$C = < subject1 > + < verb1 > + < object1 > + < CC > + < verb2 > + < object2 > \quad (4c)$$

$$C = < subject1 > + < CC > + < subject2 > + < verb2 > + < object2 > \quad (4d)$$

$$C = < subject1 > + < verb1 > + < object1 > + < CC > + < object2 > \quad (4e)$$

Each of the rules for the examples 1, 2, and 3 above can be represented as shown in 4b, 4c, and 4e. where S is either a simple sentence or an independent clause, C is a compound sentence and CC is the conjunction.

3.2 Complex Sentences

A similar pattern like above is not found in complex sentences as they comprise of a dependent clause and subordinate conjunction. So here the method required to split them is different. The dataset used in the research comprised of the following complex sentences of the pattern as given:

- (1) Sentences that have subordinating conjunction 'that' followed by a verb.
- (2) Sentences having multiple commas which express more than one thought about the same noun with the subordinating conjunction followed by a verb.
- (3) Sentences having subordinate conjunction like 'Because', 'As' in the beginning or middle of the sentence with a comma .
- (4) Sentences having more than one dependent clauses.

4. Methodology

The methodology exploits the grammar rules to segregate the sentences and then convert them into simple sentences. The dataset comprises of compound and complex sentences. Both of these sentences need to be converted to simple sentences which can be done by seeing the pattern in each and split them to simple sentences. The syntax of the compound and complex sentences is based on certain patterns and rules. Identifying the patterns can help to split the sentences into simple.

4.1 Algorithm for Compound Sentences

Input: Text 'T' of n sentences.

Output: Set of Simple sentences from compound sentences.

Begin: $T = S_1, S_2, \dots, S_n$ where S_1, S_2, \dots, S_n are the input sentences of T.

For each sentence S_j ($j=1:j<n;j++$)

Step 1: Identify if the sentence is a compound sentence. If not go to Step 6.

Step 2: Check if the removal of conjunctions or semicolon is sufficient to generate two simple sentences. If true go to Step 5 else go to step 3.

Step 3: Identify the missing components of the second independent clause before/after the conjunction.

Step 4: Add the corresponding missing components from the first independent clause to the second independent clause. In the case of the conjunction 'nor' change the structure of the second independent clause.

Step 5: Split the clauses to form simple sentences.

Step 6: Write to the output file.

End For

End

Using this pattern the following steps are carried out to extract simple sentences from the compound sentences.

- (1) Identifying the pattern of the compound sentence, **dropping the conjunction** gives the equation 5a, 5b, 5c and 5d from the corresponding equations 4b, 4c, 4d and 4e.

$$C = S + S \quad (5a)$$

Not dropping the conjunction:
These are the possible forms of the declarative sentence "before" drop the conjunction

$$C = S + S - < subject2 > \quad (5b)$$

$$C = S - < verb1 > - < object1 > + S \quad (5c)$$

$$C = S + S - < subject2 > - < verb2 > \quad (5d)$$


Explain each equation


- (2) **After**, Split the sentences based on the above equation we get two sentences S1 and S2 as 6a and 6b.

$$S1 \rightarrow < \text{subject1} > + < \text{verb1} > + < \text{object1} > \quad (6a)$$

$$S2 \text{ or } S2 \rightarrow < \text{subject2} > \text{ or } S2 \rightarrow < \text{subject2} > - < \text{verb2} > \quad (6b)$$

- (3) Check the following conditions to form the correct sentence from the two splitted sentences
- If $S2 = < \text{subject2} > + < \text{verb2} > + < \text{object2} >$ then output S1 is the first simple sentence and S2 is the second simple sentence.
 - If $S2 = < \text{verb2} > + < \text{object2} >$ then output S1 is the first simple sentence and $S2 = < \text{subject1} > + < \text{verb2} > + < \text{object2} >$ is the second simple sentence.
 - If $S2 = < \text{object2} >$ then output S1 is the first simple sentence and $S2 = < \text{subject1} > + < \text{verb1} > + < \text{object2} >$ is the second simple sentence.
 - If $S1 = < \text{subject1} >$ while $S2 = < \text{subject2} > + < \text{verb2} > + < \text{object2} >$ then output S1 = $< \text{subject1} > + < \text{verb2} > + < \text{object2} >$ while S2 is the second simple sentence.

The steps followed for each of the given types of a complex sentence is  below:

- (1) Sentences of type 1 are solved in the following way: 

- Sentences are such that the noun in the sentence before 'that' is referred to and is described more about it after the verb which comes after 'that'.
- For such sentences the closest noun to 'that' is used as a reference point
- Sentences are splitted in such a way that S1 is the first sentence until the reference point d. The second sentence S2 is the sentence with the removal of 'that' replaced by the noun

- (2) Sentences of type 2 are solved in the following way:

- Here such sentences have one dependent phrase within comma and can have certain stopwords like generally, etc.
- In such cases, the stopwords are removed and the dependent phrase is extracted from the sentences.
- Then sentences before the dependent phrase are splitted as the first simple sentence, the dependent phrase becomes the second sentence ~~while the sentence followed after the dependent phrase is the third simple sentence~~

- (3) Sentences of type 3 are solved in the following way:

- These sentences would be such as containing subordinates like 'because', 'however', 'although', 'as', 'after', etc.
- Such sentences have either the independent clause followed by a dependent clause or vice versa separated by comma
- First, the conjunctions and comma are removed then the splitting of sentences is done.
- The first sentence is the independent clause while the second sentence is the next dependent clause.

- (4) Sentences of type 4 are solved in the following way:

- Sentences belonging to this type have more than one dependent clauses.
- So the sentences are splitted into different clauses and then the main subject is appended to each clause to make it a complete sentence.
- A dictionary is created to change the tense of the verb according to the sentence.
- Accordingly based on the changed tense of the verb and the main subject appended with the dependent clause, the sentences are converted to simple.

5. Discussion and Analysis of Results

This section gives the test cases of compound sentences which are converted to simple as follows:

- (1) As the method deals with the basic form of the sentence i.e. having subject, verb, and object, any other meaning about the sentence is preserved after splitting.
 - E.g. We have never been to Asia, nor have we visited Africa
 - With the use of the above steps for compound sentences the sentence gets split into S1: We have never been to Asia and S2: We have never visited Africa
 - Hence the meaning is preserved in the sentences after splitting
- (2) The steps provided for compound sentences gives a robust solution for sentences of the type 'He likes to run and swim'.
 - Instead of splitting the first independent clause into subject, verb, and object, the first clause is used entirely, then the verb and words after the verb are combined to form the second sentence.
 - This preserves the infinitives and the other adverbs and adjectives that might precede the verb.
 - So the sentences split into S1: He likes to run and S2: He likes to swim.
 - In this case, the verb is 'run' and 'swim' with infinitives 'likes to' which is appended with the subject to form the two simple sentences.


The steps used in the algorithm have paved a way to convert the compound sentences into simple sentences. With the pattern-based rule system, the compound sentences get splitted into accurate simple sentences.

This section provides the details of the test cases of complex sentences which are converted to simple as follows:

- (1) The complex sentences of the type: 'He bought the shoes that were in the shop window. They were pretty'.
 - This kind of sentences is of type 1 for which the steps gives the simple sentences as S1: He bought the shoes. S2: The shoes were in the shop window and S3: They were pretty.
- (2) The complex sentences of the type: 'Generally, swap space is allocated a chunk of the disk, separate from the file system, so its use is as fast as possible'.
 - The above sentence is of type 2 wherein the steps split the sentence as S1: The swap space is allocated a chunk of the disk, S2: It is separate from the file system and S3: Its use is as fast as possible.
- (3) The complex sentence of the type: 'Because my coffee was too cold, I heated it in the microwave'.
 - The sentence is of type 3 wherein the sentence is split into S1: My coffee was too cold and S2: I heated it in the microwave
- (4) The complex sentence of the type: 'Artificial intelligence research has been necessarily cross-disciplinary drawing on areas of expertise such as applied mathematics, thus emerging as the latest technology'.
 - The sentence is of type 4 wherein the sentence is split into S1: Artificial intelligence research has been necessarily cross-disciplinary, S2: Artificial intelligence research drew

Table 1. Summary of the outputs of compound sentences


	Sentence
Before	I like tea and he likes coffee.
After	<i>I like tea. He likes coffee.</i>
Before	Mary never wrote the letter, nor did she call him.
After	<i>Mary never wrote the letter. She didn't call him.</i>
Before	Mary ran fast, but she couldn't catch John.
After	<i>Mary ran fast. She couldn't catch John.</i>
Before	I have known him for a long time, yet I have never understood him.
After	<i>I have known him for a long time. I have never understood him.</i>
Before	Cathy and John visited us on Thanksgiving.
After	<i>Cathy visited us on Thanksgiving. John visited us on Thanksgiving.</i>

on areas of expertise such as applied  matics and S3: Artificial intelligence research emerged as the latest technology.

The approach is implemented using Python and its libraries of NLTK (Natural Language Toolkit) and SPACY. To check the correctness of the algorithm, we have collected different compound and complex sentences from openly available online sources. ~~There is no standard for computing the performance of such systems so we have taken into account the judgments of three linguistic experts to validate the correctness of simplified sentences. The average of their judgments is considered as the accuracy of the system.~~

Compound Sentences: The dataset comprises of 166 declarative compound sentences which had 115 sentences with conjunctions while rest 51 were those having a semicolon. Among the 115 sentences 50 sentences are using 'and', 8 using 'for', 7 using 'nor', 28 using 'but', 7 using 'or', 7 using 'yet' and 8 sentences using 'so' as conjunctions. The 115 sentences are inclusive of sentences comprising of multiple conjunctions i.e. sentences using more than one conjunctions.

The algorithm was run on the entire dataset and generated 348 sentences. Out of 348 sentences, 7 sentences having multiple conjunctions were not splitted so were not converted to simple sentences. According to the human evaluators from the remaining 341 splitted sentences 334, 336 and 330 were accurate simple sentences. Therefore the system's accuracy is 97.75%. Table 3 gives the statistical details of the outputs for compound sentences. Table 1 summarizes the example outputs for different sentences. Table 2 provides the example outputs for the sentences whose meanings were not preserved after the split.

The drawback of our approach with compound sentences  cannot split sentences having multiple conjunctions and sentences having semicolon along with the conjunction. The algorithm splits sentences but for some sentences, the meanings of the splitted sentences are not preserved as seen in the 3rd and 4th sentence of table 2. This is due to the structure of the original sentence which can be assured only by human intervention. The usage of sentences involving an adjective along with the object also causes an incorrect split as seen in the 1st and 2nd sentence of table 2.

Complex Sentences: The dataset comprises of 102 declarative complex sentences among which there are 72 sentences belong to type 3 wherein the subordinate conjunction can be in the middle or start of the sentence and 10 sentences each belonging to type 1, type 2 and type 4 respectively.

Table 2. Summary of the outputs of compound sentences whose meanings were not retained

	Sentence
Before	Money is a good servant but a bad master.
After	<i>Money is a good servant. Money is a good a bad master</i>
Before	I like peanut butter and jelly
After	<i>I like peanut butter. I like peanut jelly</i>
Before	I am just going outside and may be some
After	<i>I am just going outside. I may be sometimes</i>
Before	Jim and Susie both enjoy skyng.
After	<i>Jim both enjoy skyng. Susie both enjoy skyng</i>
Before	Susie could study music or drama next year.
After	<i>Susie could study music. Susie could study drama next year.</i>

Table 3. Statistical summary for compound sentences

Total	Sentences Generated	Splitted Sentences	Correct Simple Sentences (Evaluators Judgement)	Accuracy
166	348	341	Evaluator 1: 334, Evaluator 2: 336, Evaluator 3: 330	97.75 %

Table 4. Summary of the outputs of complex sentences

	Sentence
Before	The museum was very interesting, as I expected.
After	<i>The museum was very interesting. I expected.</i>
Before	India won the tournament, defeating Sri Lanka by 6 wickets in the final at Wankhede Stadium in Mumbai, thus becoming the first country to win the Cricket World Cup final on home soil.
After	<i>India won the tournament. India defeated Sri Lanka by 6 wickets in the final at Wankhede Stadium in Mumbai. India became the first country to win the Cricket World Cup final on home soil.</i>
Before	Mihir booked the ticket in the hotel that had a spa.
After	<i>Mihir booked the ticket in the hotel. The hotel had a spa.</i>
Before	However, the duties of archchancellor for Italy were generally discharged by deputy, and after the virtual separation of Italy and Germany, the title alone was retained by the elector.
After	<i>The duties of archchancellor for Italy were generally discharged by deputy. The title alone was retained by the elector.</i>

With 102 complex sentences, 204 simple sentences have been generated. Out of 204, 18 sentences were not split as they comprised of more than 1 noun. So the remaining 186 sentences were grammatically correct. As per the human evaluators among these sentences, some of them could not convey the exact meaning of the original sentence. Table 6 provides the statistical details for complex sentences. Table 4 summarizes the example outputs for different sentences. Table 5 provides the example outputs for the sentences whose meanings were not conveyed after the split.

The drawback in the case of complex sentences is it cannot split the sentences if there is more than one noun in the sentences. The approach does not retain the event occurrence or temporal situation of the original sentences after the split. As shown in table 5 the sentences are split correctly and are grammatically correct but the event that a person can be healthy and fit only if he exercises cannot be conveyed as in the first sentence. A temporal situation depicted in the original sentence cannot be conveyed in the splitted simple sentences as seen in the fourth splitted sentence of table 5. Similarly, the events of ordering a particular dish only in that restaurant in the second sentence or presence of nervousness prior of the third sentence are not conveyed in the splitted sentences.

Table 5. Summary of the outputs of complex sentences whose meanings were not conveyed

	Sentence
Before	He will be able to maintain a healthy weight if he keeps exercising.
After	<i>He will be able to maintain a healthy weight. He keeps exercising.</i>
Before	Whenever they eat at this restaurant, they order a hamburger and fries.
After	<i>They eat at this restaurant. They order a hamburger and fries.</i>
Before	Even though I was nervous about the date, I had a really great time after we started talking.
After	<i>I was nervous about the date. I had a really great time after we started talking.</i>
Before	After they finished studying, Emily and Lily went to the movies.
After	<i>They finished studying. Emily and Lily went to the movies.</i>

Table 6. Statistical summary for complex sentences

Total	Sentences Generated	Splitted Sentences	Correct Simple Sentences (Evaluators Judgement)	Accuracy
102	204	186	Evaluator 1: 169, Evaluator 2: 166 , Evaluator 3: 165	89.60 %



6. Conclusion

For extracting information in the NLP applications simple sentences are pre-requisite. The existing approaches make use of only simple sentences as input. However, the given data set or a text document could include various types of sentences requiring its simplification. Hence conversion of sentences from complex and compound form of sentences to simple sentences is essential for natural language applications such as anaphora resolution, sentiment analysis, machine translation and many more. In this direction syntactic simplification of the sentences has been proposed in this paper which not only identifies compound sentences and complex sentences but also simplifies them to simple sentences. The proposed technique identifies the patterns of the compound and complex sentences and feeds this to the algorithm which exploits this pattern to simplify the sentences. This system can be utilized to generate the simple sentences which can be input to the various automatic tools used for NLP tasks.

In this paper, a new approach to simplify sentences from both compound and complex sentences has been introduced. An extension is to make the system to handle compound sentences having multiple conjunctions and complex sentences having more than one noun. This can be achieved by exploring the patterns of such sentences in CFG. Also, another long term objective is to check the performance of the system as it depends on SPACY and NLTK libraries of Python. Future direction is targeted to generate multiple-choice questions from the simple sentences generated from our proposed technique.

References

- S. S. Yadav and S. Bojewar 2015. Survey on Automated Text Documents Summarization Tools. *International Journal for Research in Engineering Application & Management (IJREAM)*, 1.
- A. Bawakid and M. Oussalah 2011. Sentences simplification for automatic summarization. In *Proceedings of IEEE 10th International Conference of Cybernetic Intelligent Systems (CIS)*, IEEE.
- A. M. Azmi and N. I. Altamami 2018. An abstractive Arabic text summarizer with user controlled granularity. *Information Processing and Management*, 54:903–921.
- B. Backman 2003. *Building Sentence Skills: Tools for Writing the Amazing English Sentence*.
- B. Das, M. Majumdar and S. Phadikar 2018. A Novel System for Generating Simple Sentences from Complex and Compound Sentences. *International Journal of Modern Education and Computer Science*, 1:57–64. DOI: 10.5815/ijmecs.2018.01.06.
- B. F. Klimova 2013. Syntactic-semantic classification of sentences. In *2nd World Conference on Design, Arts and Education DAE*.
- C. Poornima, V. Dhanalakshmi, K. M. Anand and K. P. Soman 2011. Rule based sentence simplification for english to

- tamil machine translation system. *International Journal of Computer Applications*, 25(8):38–42.
- C. Scarton, G. Paetzold, and L. Speci** 2018. Simpa: A sentence-level simplification corpus for the public administration domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may. European Language Resources Association (ELRA).
- D. Jurafsky and H. James** 1996. *Speech and language processing an introduction to natural language processing, computational linguistics, and speech*.
- D. R. Ch and S. K. Saha** 2018. Automatic Multiple Choice Question Generation from Text : A Survey. *IEEE transactions on Learning Technologies*.
- D. Vickrey and D. Koller** 2008. Sentence simplification for semantic role labeling. In *Proceedings of the Association for Computational Linguistics(ACL)*.
- F. A. Tarouti, J. Kalati, and C. McGrory** 2015. Sentence simplification for question generation. In *Proceedings of the International Conference on Computing and Communication Systems*.
- F. Obrecht** 1999. *Minimum Essentials of English*. Barron's Educational Series.
- G. Lutz and D. Stevenson** 2005. *The Writer's Digest Grammar Desk Reference*.
<https://examples.yourdictionary.com/> 2020. Complex Sentences.
- J. D. Belder and M. F. Moens** 2010. Text Simplification for children. In *Proceedings of the SIGIR Workshop on Accessible Search Systems*.
- J. Frost** 2020. English Grammar.
- J. Sevastopoulos** 2019. English Grammar2.
- L. Brouwers, D. Bernhard, A. L. Ligozat, and T. Francois** 2014. Syntactic sentence simplification for French. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*.
- M. Heilman and N. A. Smith** 2010. Extracting simplified statements for factual question generation. In *Proceedings of QG2010: the Third Workshop on Question Generation*.
- M. Majumder and S. K. Saha** 2015. A system for generating multiple choice questions: With a novel approach for sentence selection. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications(IJNLP)*.
- M. Miwa, R. Satre, Y. Miyao and J. I. Tsujii** 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Association for Computational Linguistics)(COLING)*.
- O. Biran, S. Brody, and N. Elhadad** 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies(ACL)*.
- R. Chandrasekar, C. Doran, and B. Srinivas** 1996. Motivations and methods for text simplification. In *Proceedings of the 16th Conference on Computational linguistics (Association for Computational Linguistics)(COLING)*.
- R. Sukthankar, S.Poria, E. Cambria, and R. Thirunavukarasu** 2018. Anaphora and Coreference Resolution: A Review. In *arXiv:1805.11824*, Available: : <https://arxiv.org/abs/1805.11824>.
- S. Jonnalagadda and G. Gonzalez** 2010. Biosimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction. In *Proceedings of AMIA*.
- S. K. D. Nikita and S. K. Sharma** 2015. Detection of complex sentences in Punjabi language using CRF. *Journal of Innovation in Electronics and Communication Engineering*, 5:42–46.
- S. Narayan, C. Gardent, S. B. Cohen and A. Shimorina** 2016. Unsupervised Sentence Simplification Using Deep Semantics. In *Proceedings of INLG*.
- S. Narayan, C. Gardent, S. B. Cohen, and A. Shimorina** 2017. Split and Rephrase. In *Proceedings in EMNLP*.
- S. R. Savanur and Dr. R. Sumati** 2018. Feature Based Sentiment Analysis of Compound Sentences. *International Journal of Modern Education and Computer Science*, 1:57–64.
- S. Wubben, A. V. D. Bosch, and E. Krahmer** 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics(ACL)*.
- Siddharthan, A.** 2003. *Syntactic simplification and text cohesion*. PhD thesis, University of Cambridge.
- T. B. McArthur and F. McArthur** 1999. *The Oxford Companion to the English Language, Oxford Companions Series*. Oxford University Press.
- T. Chen, R. Xu, Y. He and X. Wang** 2017. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems With Applications*, 72:221–230.
- T. P. Klammer, R. M. Shultz, and A. D. Volpe** 2007. *Analyzing English Grammar*.
- U. C. Jaiswal, R. Kumar, and S. Chandra** 2009. A Structure based Computer Grammar To understand Simpel and Compound English sentences. In *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3'09)*.
- V. Nq** 2015. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics(ACL)*.

- W. Coster and D. Kauchak** 2011. Learning to simplify sentences using wikipedia. In *In Proceedings of Monolingual Text-To-Text Generation*.
- Y. Peng, C. O. Tudor, M. Torii, C. H. Wu, and K. Vijay-Shanker** 2012. iSimp: A sentence simplification system for biomedical text. In *Proceedings of the Bioinformatics and Biomedicine (BIBM)IEEE*.
- Z. Zhu, D. Bernhard, and I. Gurevych** 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Association for Computational Linguistics(COLING))*.