

# **“Automatic Story Generator”**

**A**

## ***Project Report***

*submitted in partial fulfillment of the  
requirements for the award of the degree of*

## **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

**by**

<b>Name</b>	<b>Roll No.</b>
<b>Sanyam Jain</b>	<b>R114215041</b>
<b>Sonaal Kalra</b>	<b>R114215045</b>
<b>Shivani Sharma</b>	<b>R114215057</b>

*Under the guidance of*

**Dr. Hitesh Kumar Sharma**

**Ass. Professor**



**Department of Analytics  
School of Computer Science and Engineering  
University of Petroleum & Energy Studies  
Bidholi, Via Prem Nagar, Dehradun, UK  
November – 2017**



The innovation driven  
**E-School**

## **CANDIDATE'S DECLARATION**

I/We hereby certify that the project work entitled “ **Automatic Story Generator**” in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Ecommerce Retail and Automation) and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August, 2017** to **November, 2017** under the supervision of **Dr. Hitesh Kumar Sharma**.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

**(Sanyam Jain  
Shivani Sharma  
Sonaal Kalra)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Nov. 8, 2017

**(Dr. Hitesh Kumar Sharma)**  
Project Guide

**Dr. T.P. Singh**  
Program Head – Btech CSE ECRA 2015-19  
Center for Information Technology  
University of Petroleum & Energy Studies  
Dehradun – 248 001 (Uttarakhand)

## 1. ABSTRACT

“*Automatic Story Generator*” is a program written in C programming language that generates random sentences limiting keywords in the WordBase. WordBase is corpus of words that are stored in arrays of sentence makers. The tool provides a vast possibility to add as many words and generate as many sentences. The main task is to achieve meaningful sentence out of all the random sentences. The Randomness can be categorized as sort of spinning the text.

The term story generator algorithms (SGAs) refer to computational procedures resulting in an artifact that can be considered a story. In the field of Artificial Intelligence (AI), the automated generation of stories has been a subject of research for over fifty years. An algorithm is understood as a set of instructions that, when applied to a given input, produces an output. In the present context, the desired output is a story. The underlying concept of “story” in SGAs is functional and does not imply any aesthetic notion. This is important because it sets the context for evaluation of generated stories, for which having a surface realization as a readable and appealing text is not necessarily a core issue

## 2. PROBLEM STATEMENT

1. Thinking about some stories and mind but not able to make or complete it
2. Making any random content or write any content just by having keywords is not easy
3. Writing papers, genre specific content, Narrating, Natural language generation are major aspects of AI.

## 3. LITERATURE REVIEW

REFERENCE	AUTHOR	KEY FINDINGS	COMMENTS
[1]	Y Shim ,M KIM	generate consistent stories using autonomous characters have multi-level goals: viewer goals, plot goals, and character goals A Model for Generating Consistent Episodes	According to this to construct an interactive narration, it use multi-level goals: viewer goals, plot goals, and character goals. The character goal is used to make believable and emotional characters. The plot goal is used to generate a consistent story. The viewer goal is used to interact with users.

[2]	CD Newell, Md wood	A method and system for automatically creating an image product based on assets stored in a user database obtaining a plurality of digital media files associated with an event; automatically classifying the event based on analyzing said plurality of digital media files;	A number of stored digital media files are analyzed to determine their semantic relationship to an event and are classified according to requirements and semantic rules for generating the image product.
[3]	Neil McIntyre and Mirella Lapata	A data-driven approach for generating short children's stories that does not require extensive manual involvement. System follows a generate-and-and-rank approach Stories generated by the random, deterministic, and rank-based systems.	It proposed a novel method to computational story telling. This approach has three key features. Firstly, story plot is created dynamically. Secondly, generator realizes the various components of the generation pipeline. Thirdly, it generate and store multiple stories efficiently in a tree data structure.
[4]	P Gervas	Only narrate the main events of the plot System operates with a representation in Description Logics, combining stored fabulas with the narrative knowledge implemented in a domain-specific ontology. utility of the system in terms of quality and originality of the generated artifact	The aim of our project is to generate creatively new basic stories. This deals with how to create more useful and new at the same time. Content production is being done using theories in the field of Narratology
[5]	A Hong, C Solis, JT Siy, E Tabirao	Natural language generation, story generation, story planning operators, story tree, semantic ontology Picture Books is an automated story generation system	THE PICTURE BOOKS SYSTEM: Picture Books is an automated story generation system intended for children age four to six. It derives the story elements from a given input picture with components selected by the user from a library of background images, character stickers and object stickers

## 4. OBJECTIVE

The objective of this project is to generate the random content automatically. The Program have some dictionary keywords stored in array. Arrays passes through Random function and spinning functions produces some sentences with concatenation of each word.

## 5. METHODS AND ALGORITHMS

1. Program will be trained for the keywords, subjects, objects and all real world entities.
2. Those keywords are coded such that it makes fresh, real and original content every time.
3. Several algorithms for making random generations, string related algorithms, time calculation algorithms etc will be used for making our program more efficient each time of use.
4. Program have some dictionary keywords stored in array. Arrays passes through Random function and spinning functions produces some sentences with concatenation of each word.
5. One more filter make sure that the sentence going to generate is actually meaningful.
6. More basic (Uses naïve approach of English sentence and grammar algorithms )

### 6.1 Pseudocode: generatesentence()

```
char* generateSentence() {
    char* sentence = calloc((SEN_LEN+1), sizeof(char));
    const char* ARTICLES[] = {"Microsoft", "Google", "Facebook", "Twitter", "Amazo
const char* NOUNS[] = {"published", "revealed", "announces", "developed", "prep
const char* VERBS[] = {"latest", "newest", "biggest", "advanced", "fastest", "co
const char* PREPOSITIONS[] = {"software", "app", "technology", "service", "prod
const int ARTICLES_SIZE = sizeof(ARTICLES)/sizeof(ARTICLES[0]);
const int NOUNS_SIZE = sizeof(NOUNS)/sizeof(NOUNS[0]);
const int VERBS_SIZE = sizeof(VERBS)/sizeof(VERBS[0]);
const int PREPOSITIONS_SIZE = sizeof(PREPOSITIONS)/sizeof(PREPOSITIONS[0]);
    //Build Sentence
    strcat(sentence, ARTICLES[rand()%ARTICLES_SIZE]);

    strcat(sentence, " ");
    strcat(sentence, NOUNS[rand()%NOUNS_SIZE]);

    strcat(sentence, " ");
    strcat(sentence, VERBS[rand()%VERBS_SIZE]);

    strcat(sentence, " ");
    strcat(sentence, PREPOSITIONS[rand()%PREPOSITIONS_SIZE]);
}
```

## 6.2 Pseudocode: main()

```
char* sentence = generateSentence();
printf("%s.", sentence);
free(sentence);
char* sent = generateSent();
printf("%s.", sent);
free(sentence);
char* sente = generateSente();
printf("%s.\n", sent);
free(sentence);
```

## 6.3 Functions Used:

The C library function `void *calloc(size_t nitems, size_t size)` allocates the requested memory and returns a pointer to it.

*Sizeof* is a much used in the C programming language. It is a compile time unary operator which can be used to compute the size of its operand. The result of `sizeof` is of unsigned integral type which is usually denoted by `size_t`. `sizeof` can be applied to any data-type, including primitive types such as integer and floating-point types, pointer types, or compound datatypes such as Structure, union etc.

The C library function `int rand(void)` returns a pseudo-random number in the range of 0 to `RAND_MAX`.

## 7.1 DFD DIAGRAM

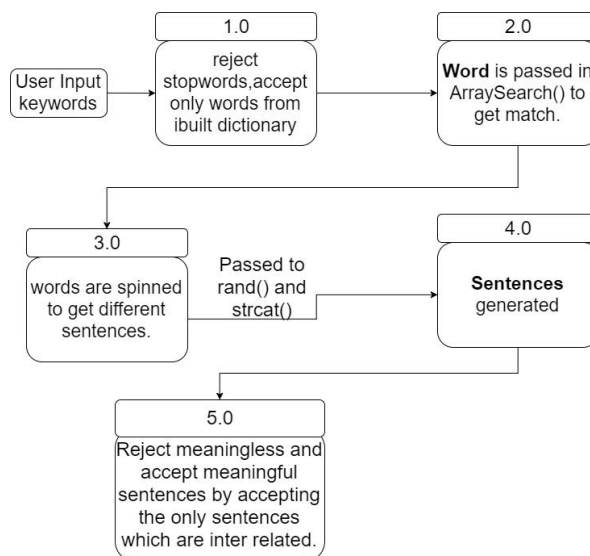
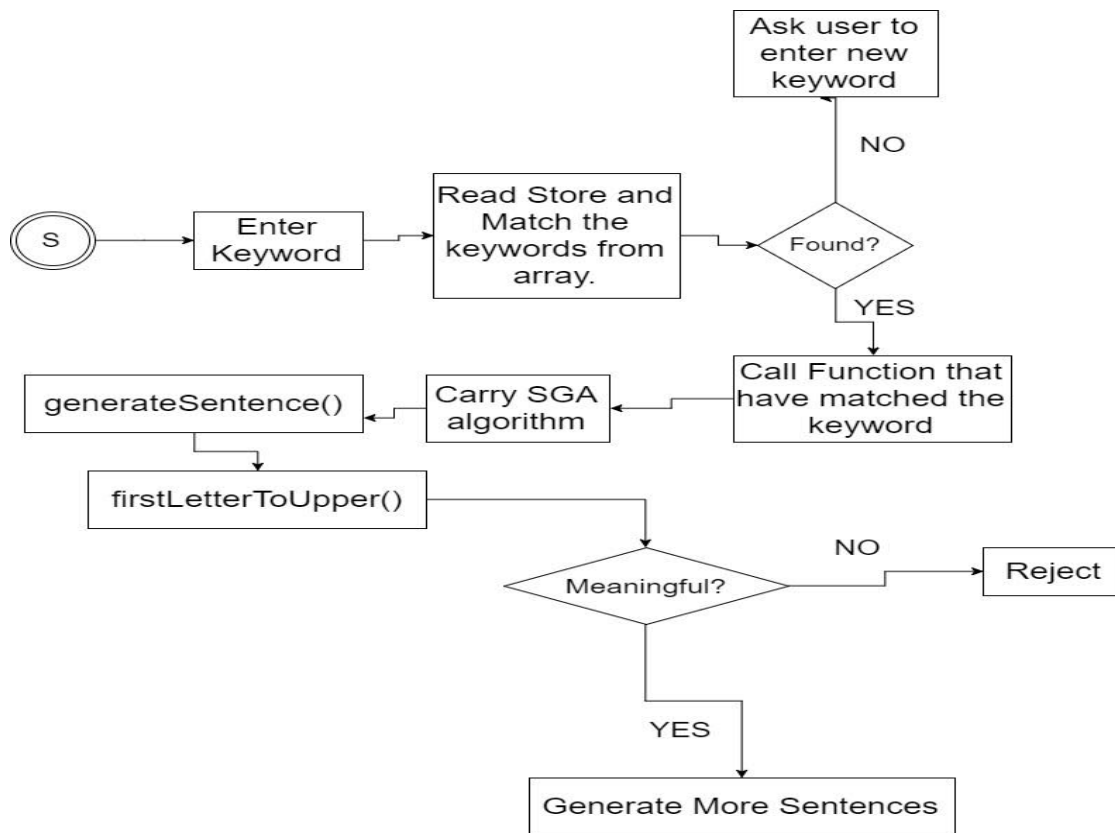


Figure 1. BROAD VIEW  
OF ASG

## 7.2 FLOWCHART



### GenerateSentence() Method

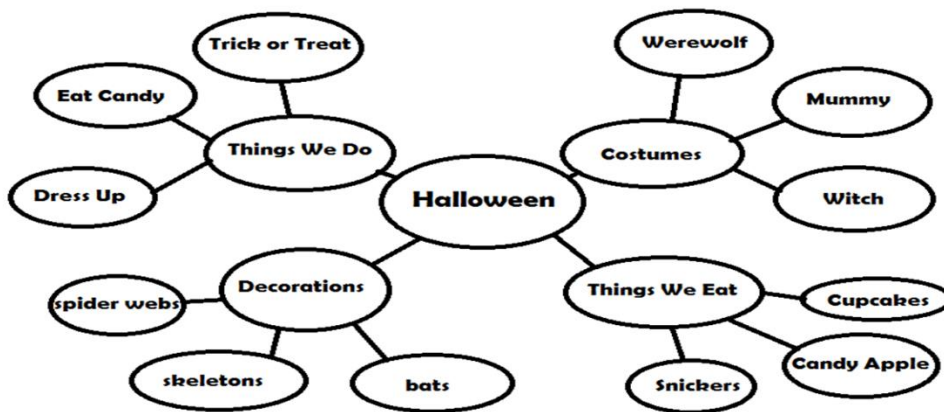
```
strcat(sentence,Articles)
    " "
strcat(sentence,Nouns)
    " "
strcat(sentence,Verbs)
    " "
strcat(sentence,Prepositions)
    " "
strcat(sentence,Finalizers)
    " "
return sentence;
```

## 8. OBJECTIVE ACHIEVED

- Program is compiled with best of compiler GNU C or GCC and gedit is used
- Analysed with different compilers to check which is best. The compiler with least compilation time will be chosen as best. The compilation time can be traced with certain time.h library functions.
- Effective Procedural Code
- Manages System Resources. Ubuntu 16.04 (Linux) is well optimized for native C

S.M.A.R.T. project objectives

1. **SPECIFIC:** Writing any random content is not easy task, so as to generate random sentences collectively stories.
2. **MEASURABLE:** The distinguished importance of the ASG is that it creates random stories within itself and output is more accurate in terms of meaning of actual English sentence generated. The desired random() function was coded in mid September which generated random output of meaningless sentences. Then late number we Applied Grammar Algorithm. Failing in that resulted to make our own functions similar to SGA/GA to make meaningful sentences.
3. **Attainable/Achievable:** The goal is to achieve a collection of inter related meaningful sentences. We have reach to the meaningful sentences generations. The inter relativity is achieved by functionalizing each short SGAs.(Having same array stores, interrelated keywords, which should make a net like structure of words



4. **Relevancy:** The relevancy of each function which mark its importance to generate each meaningful sentence proves the relevancy of the program. The program is relevant in real life scenarios as :

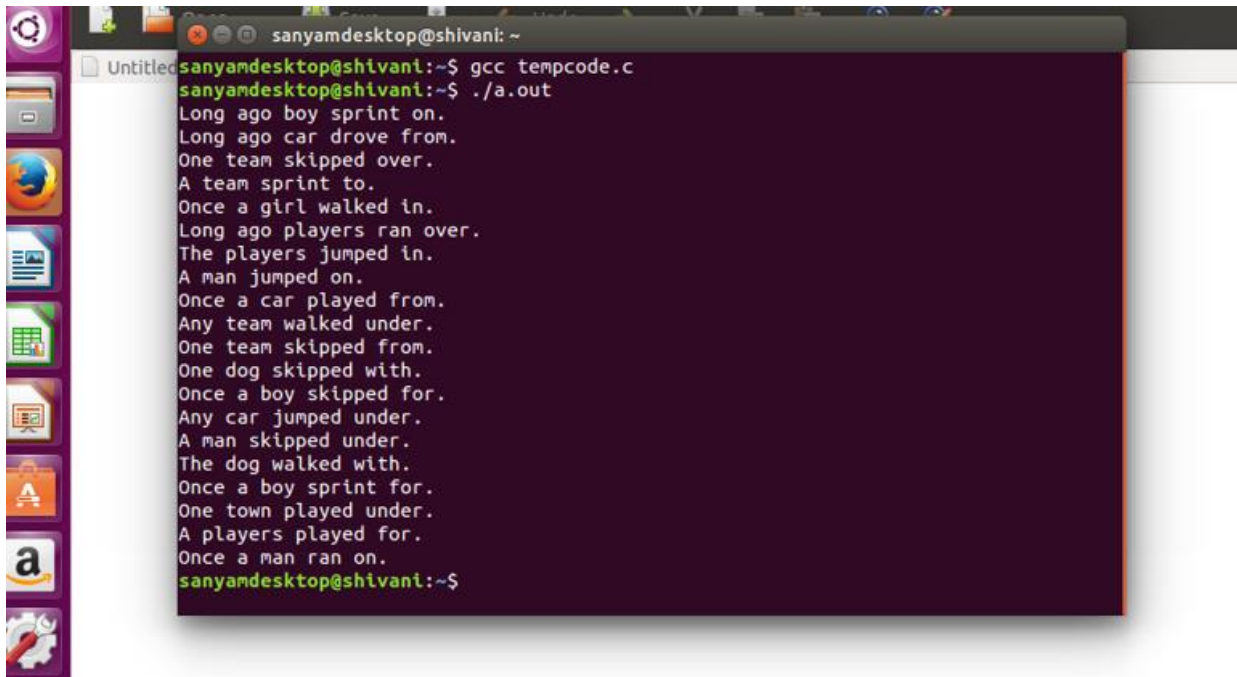
- A popular application of SGA is TALESPIN used late 70's using lisp language to generate stories of cartoon characters.
- MINSTREL (Turner 1993) was a computer program that told stories about King Arthur and his Knights of the Round Table.
- 5. **Time-Bound :** The complete project should end up before end semesters



## 9. RESULTS

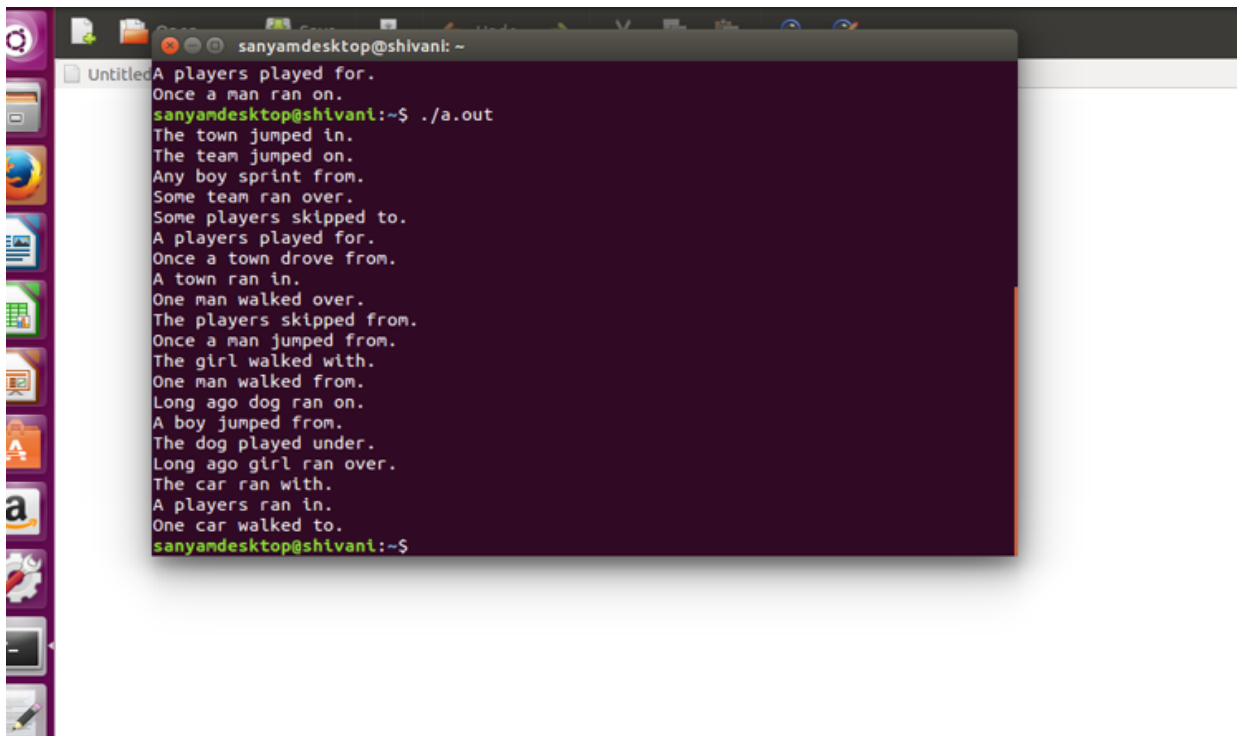
Here are the sample outputs of the automatic story generator

1: Random stories without meaning



A terminal window titled 'sanyamdesktop@shivani: ~' showing the execution of a C program. The user enters 'gcc tempcode.c' and './a.out'. The program outputs a list of 25 random sentences. The window has a dark purple background and a sidebar with application icons on the left.

```
sanyamdesktop@shivani: ~  
sanyamdesktop@shivani:~$ gcc tempcode.c  
sanyamdesktop@shivani:~$ ./a.out  
Long ago boy sprint on.  
Long ago car drove from.  
One team skipped over.  
A team sprint to.  
Once a girl walked in.  
Long ago players ran over.  
The players jumped in.  
A man jumped on.  
Once a car played from.  
Any team walked under.  
One team skipped from.  
One dog skipped with.  
Once a boy skipped for.  
Any car jumped under.  
A man skipped under.  
The dog walked with.  
Once a boy sprint for.  
One town played under.  
A players played for.  
Once a man ran on.  
sanyamdesktop@shivani:~$
```

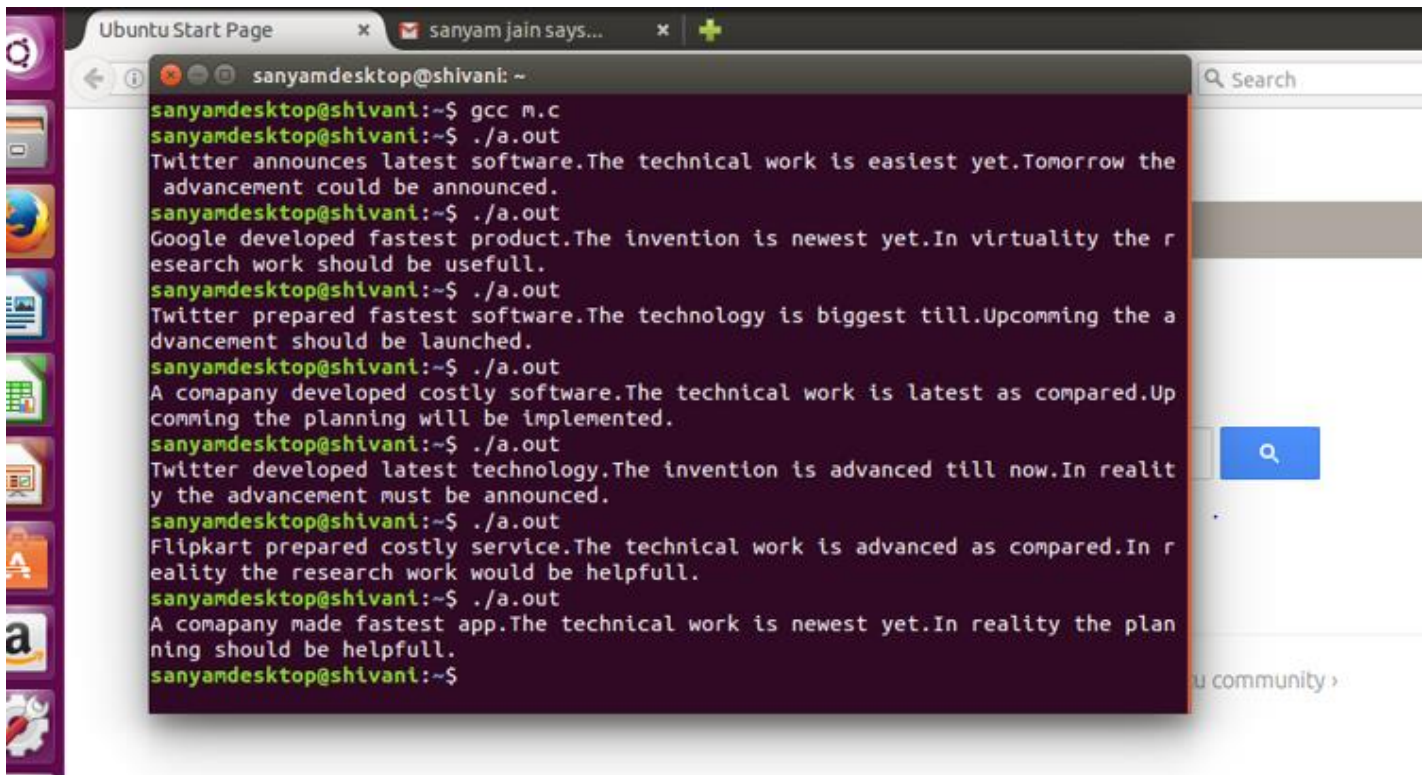


A terminal window titled 'sanyamdesktop@shivani: ~' showing the execution of a C program. The user enters './a.out'. The program outputs a list of 25 random sentences. The window has a dark purple background and a sidebar with application icons on the left.

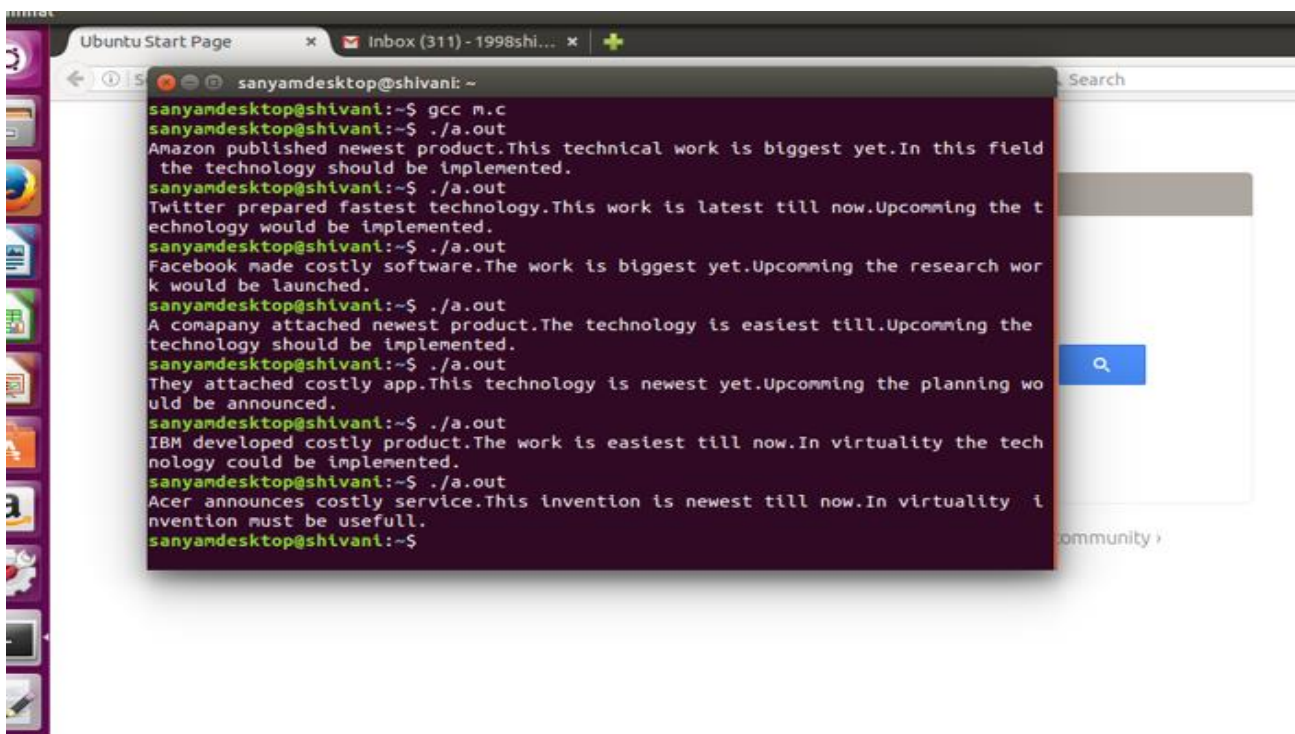
```
sanyamdesktop@shivani: ~  
A players played for.  
Once a man ran on.  
sanyamdesktop@shivani:~$ ./a.out  
The town jumped in.  
The team jumped on.  
Any boy sprint from.  
Some team ran over.  
Some players skipped to.  
A players played for.  
Once a town drove from.  
A town ran in.  
One man walked over.  
The players skipped from.  
Once a man jumped from.  
The girl walked with.  
One man walked from.  
Long ago dog ran on.  
A boy jumped from.  
The dog played under.  
Long ago girl ran over.  
The car ran with.  
A players ran in.  
One car walked to.  
sanyamdesktop@shivani:~$
```

Latest Outputs with meaningful sentence:-

2. Technical based:-



```
sanyamdesktop@shivani: ~  
sanyamdesktop@shivani:~$ gcc m.c  
sanyamdesktop@shivani:~$ ./a.out  
Twitter announces latest software.The technical work is easiest yet.Tomorrow the  
advancement could be announced.  
sanyamdesktop@shivani:~$ ./a.out  
Google developed fastest product.The invention is newest yet.In virtuality the r  
esearch work should be usefull.  
sanyamdesktop@shivani:~$ ./a.out  
Twitter prepared fastest software.The technology is biggest till.Upcomming the a  
dvancement should be launched.  
sanyamdesktop@shivani:~$ ./a.out  
A comapany developed costly software.The technical work is latest as compared.Up  
comming the planning will be implemented.  
sanyamdesktop@shivani:~$ ./a.out  
Twitter developed latest technology.The invention is advanced till now.In realit  
y the advancement must be announced.  
sanyamdesktop@shivani:~$ ./a.out  
Flipkart prepared costly service.The technical work is advanced as compared.In r  
eality the research work would be helpfull.  
sanyamdesktop@shivani:~$ ./a.out  
A comapany made fastest app.The technical work is newest yet.In reality the plan  
ning should be helpfull.  
sanyamdesktop@shivani:~$
```



```
sanyamdesktop@shivani: ~  
sanyamdesktop@shivani:~$ gcc m.c  
sanyamdesktop@shivani:~$ ./a.out  
Amazon published newest product.This technical work is biggest yet.In this field  
the technology should be implemented.  
sanyamdesktop@shivani:~$ ./a.out  
Twitter prepared fastest technology.This work is latest till now.Upcomming the t  
echnology would be implemented.  
sanyamdesktop@shivani:~$ ./a.out  
Facebook made costly software.The work is biggest yet.Upcomming the research wor  
k would be launched.  
sanyamdesktop@shivani:~$ ./a.out  
A comapany attached newest product.The technology is easiest till.Upcomming the  
technology should be implemented.  
sanyamdesktop@shivani:~$ ./a.out  
They attached costly app.This technology is newest yet.Upcomming the planning wo  
uld be announced.  
sanyamdesktop@shivani:~$ ./a.out  
IBM developed costly product.The work is easiest till now.In virtuality the tech  
nology could be implemented.  
sanyamdesktop@shivani:~$ ./a.out  
Acer announces costly service.This invention is newest till now.In virtuality i  
nvention must be usefull.  
sanyamdesktop@shivani:~$
```

## 10. CONCLUSION

From the above research papers, it is concluded that:

- [1] Generate consistent stories using autonomous characters.
- [2] System for automatically creating an image product based on word base in arrays.
- [3] System follows an automatic approach to create meaningful content.
- [4] Content production is being done using theories in the field of Narratology.
- [5] By combining some functions a meaningful story is generated.

## 11. REFERENCES:

Hong, A., Solis, C., Siy, J. T., Tabirao, E., & Ong, E. (2008). Picture books: Automated story generator. *Undergraduate Thesis, De La Salle University, Manila, Philipian*

Shim, Yunju, and Minkoo Kim. "Automatic short story generator based on autonomous agents." *Intelligent Agents and Multi-Agent Systems* (2002): 561-568.

Newell, Catherine D., et al. "Automatic story creation using semantic classifiers for images and associated meta data." U.S. Patent Application No. 11/758,358.

Cua, Jeffrey, et al. "Representing story plans in SUMO." *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*. Association for Computational Linguistics, 2010.

Onuczko, Curtis, et al. "Automatic Story Generation for Computer Role-Playing Games." *AIIDE*. 2006.

[http://wikis.sub.uni-hamburg.de/lhn/index.php/Story\\_Generator\\_Algorithms](http://wikis.sub.uni-hamburg.de/lhn/index.php/Story_Generator_Algorithms)

**Approved By**

**Project Guide**

**Dr. Hitesh Kumar Sharma**

**(Department of Analytics)**

**Program Head**

**Dr. T.P. Singh**