**“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**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Sanyam Jain** | **R11425041** |
| **Sonaal Kalra** | **R114215046** |
| **Shivani Sharma** | **R114215057** |

***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**

****

**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: \_\_\_\_\_\_\_\_\_\_\_\_\_2017 **(Dr. Hitesh Kumar Sharma)**

Project Guide

**<Name of Program Head>**

Program Head - <Name of Course>

Center for Information Technology

University of Petroleum & Energy Studies

Dehradun – 248 001 (Uttarakhand)

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Name**, for all advice, encouragement and constant support he has given us through out our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, **Name**, for his great support in doing our project in **Area (like Network, Big data etc.)** at **SoCSE**.

We are also grateful to **Name, Associate Dean** and **Name** Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Sanyam Jain** | **Shivani Sharma** | **Sonaal Kalra** |  |
| **Roll No.** | **R114215041** | **R114215057** | **R114215047** |  |

**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) refers 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

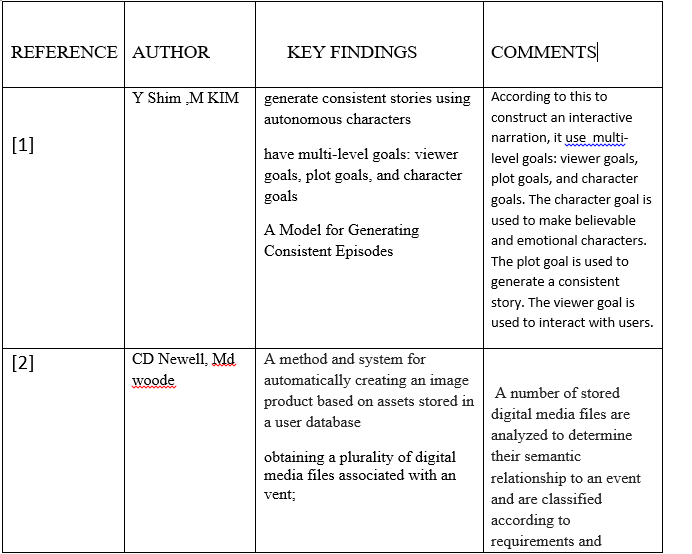
**PROBLEM STATEMENT**

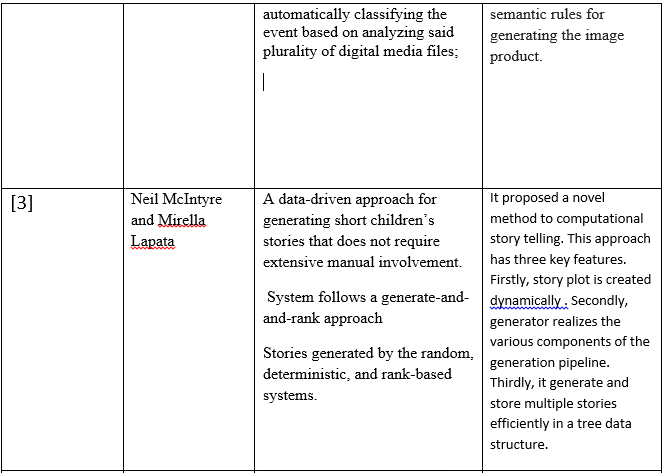
1. Thinking about some stories and mind but not able to make or complete it

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.

**LITERATURE REVIEW**





**OBJECTIVE**

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

**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.

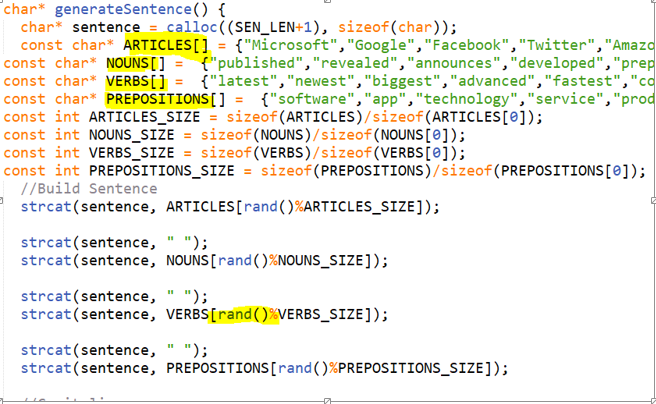
3. Several algorithms for making random generations, string related algorithms, time calculation algorithms etc will be used for making our program more efficient each time of use.

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.

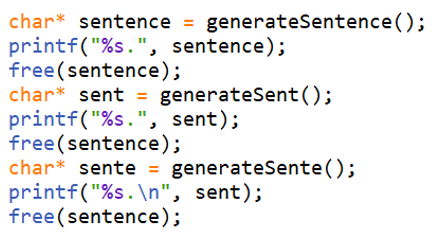
5. One more filter make sure that the sentence going to generate is actually meaningful.

6. More basic (Uses naïve approach of English sentence and grammar algorithms )

**Pseudocode: generatesentence()**



**Pseudocode: main()**

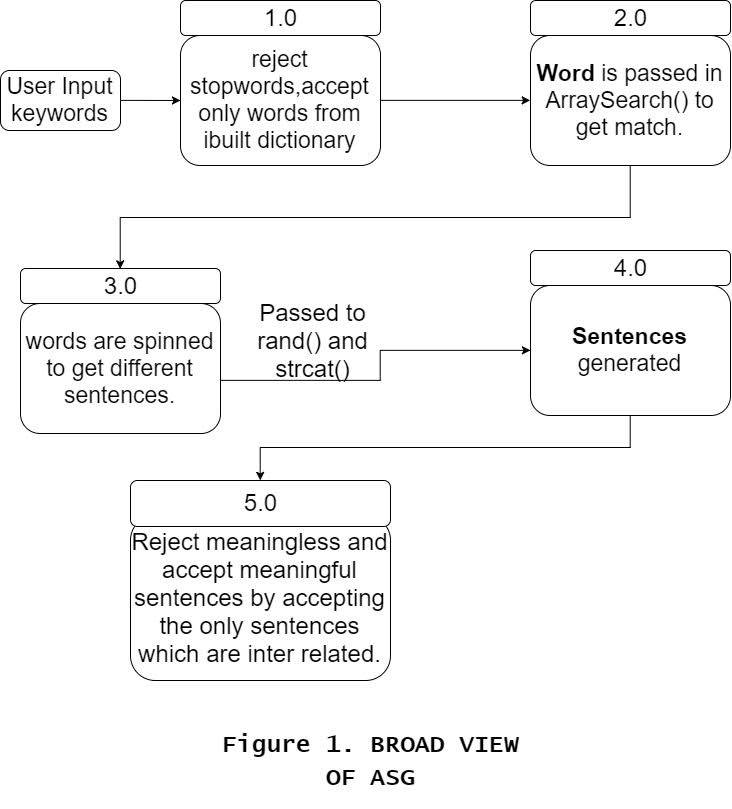


**Functions Used:**

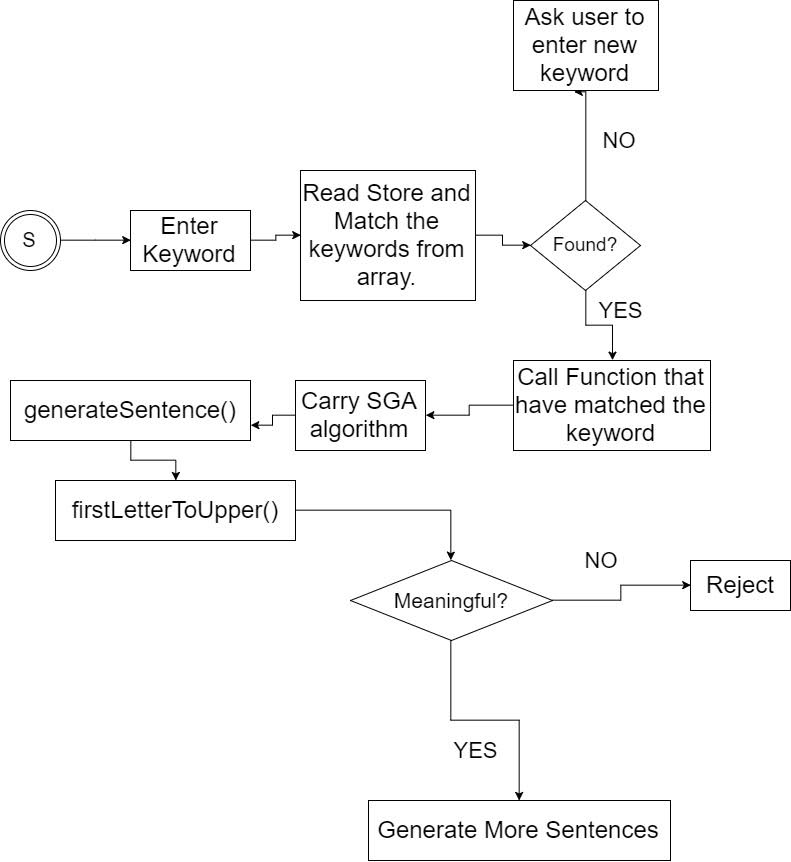
The C library function void \*calloc(size\_t nitems, size\_t size) allocates the requested memory and returns a pointer to it.

*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.

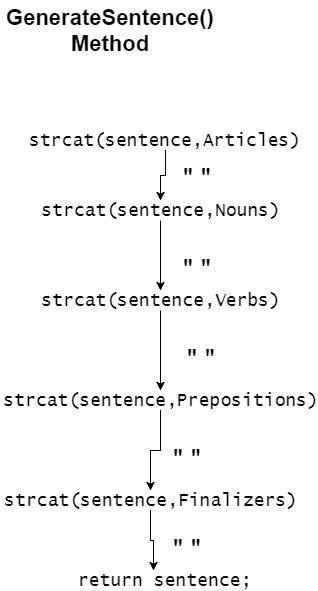
The C library function int rand(void) returns a pseudo-random number in the range of 0 to *RAND\_MAX*.

DFD DIAGRAM

**FLOWCHART**



2.

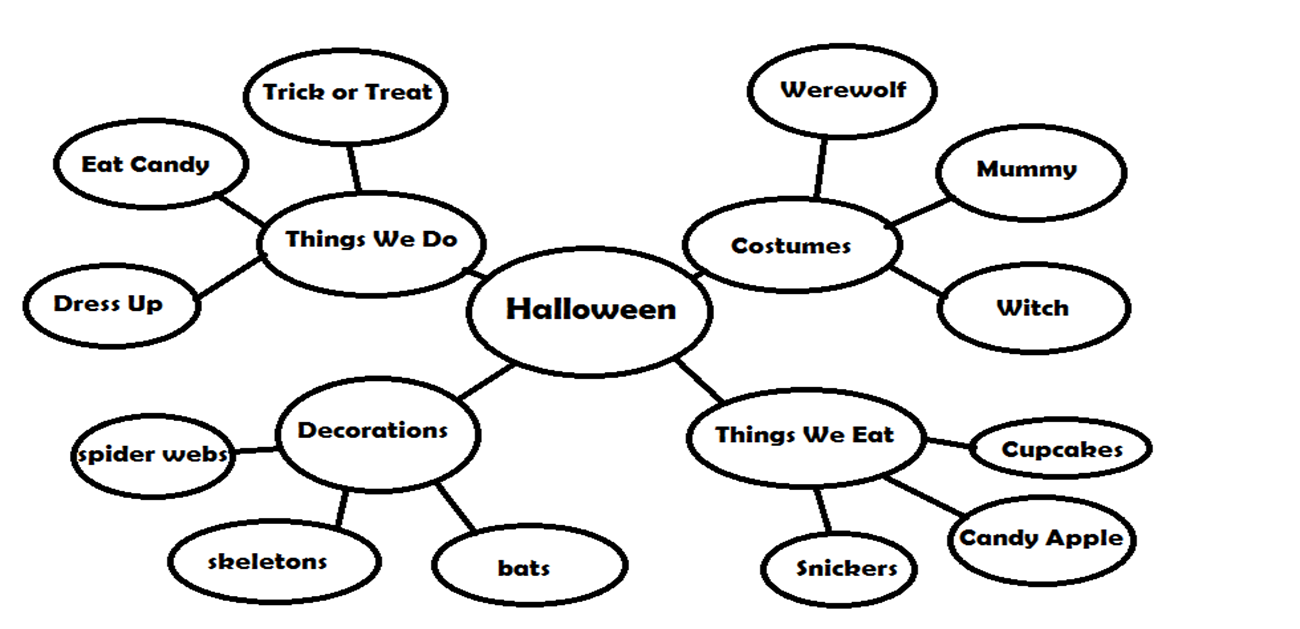


**OBJECTIVE ACHIEVED**

* 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



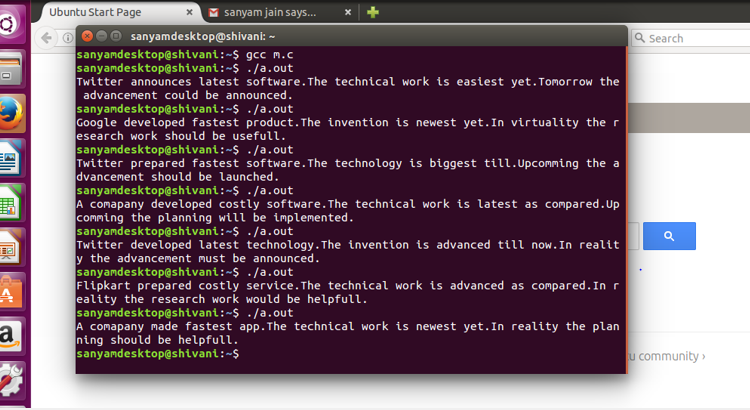
4. Relevancy: The relevancy of each function which mark its importance to generate each meaningful sentence proves the relevancy of the program. The program is relevant in real life scenarios as :

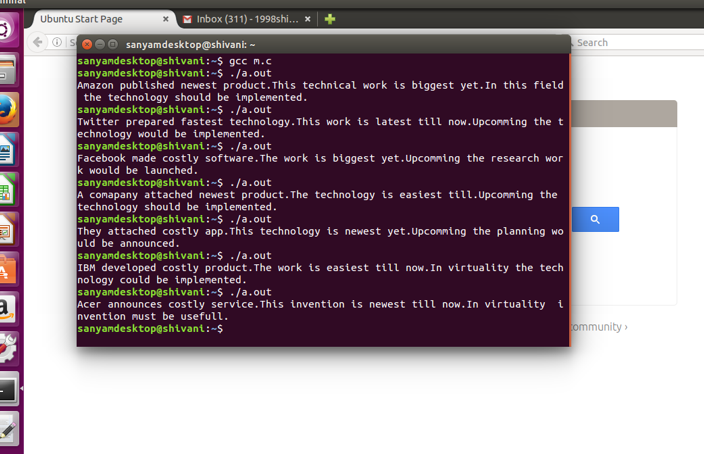
* + - A popular application of SGA is TALESPIN used late 70’s using lisp language to generate stories of cartoon characters.
    - MINSTREL (Turner 1993) was a computer program that told stories about King Arthur and his Knights of the Round Table.
* 5. Time-Bound : The complete project should end up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

**RESULTS**

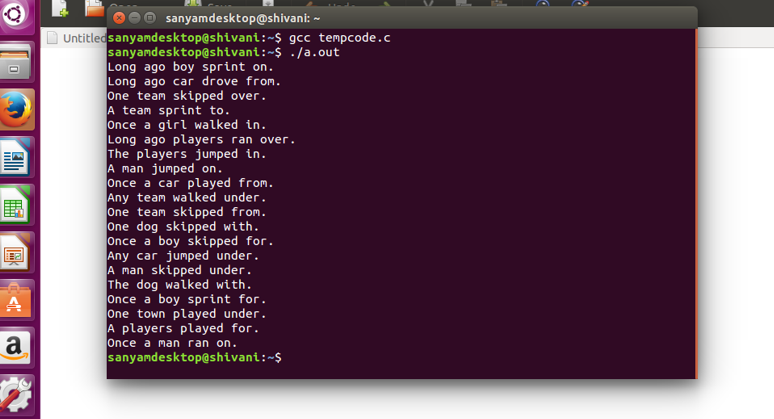
Here are the sample outputs of the automatic story generator

1

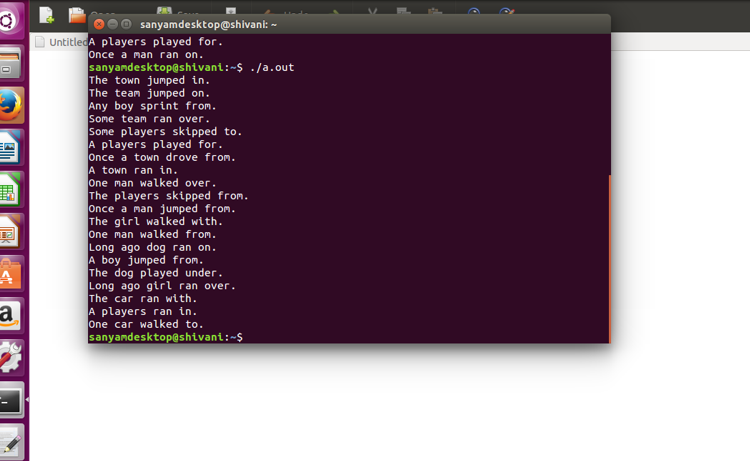


2.

3.



4



**CONCLUSION**

**REFERENCES:**

[1] 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*

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

[3] 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.

[4] 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.

[5] 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>