



## Markov Chains Text Generation for Ideation Platforms

Isaac Terngu Adom\*

Department of Computer Engineering, Benue State University, Makurdi, Nigeria

\*Corresponding author: Isaac Terngu Adom, Computer Engineering, Benue State University, Makurdi, Nigeria; E-mail: iadom@bsum.edu.ng

Received date: 31 March, 2023, Manuscript No. JCEIT-23-93819;

Editor assigned date: 03 April, 2023, PreQC No. JCEIT-23-93819 (PQ);

Reviewed date: 17 April, 2023, QC No. JCEIT-23-93819;

Revised date: 24 May, 2023, Manuscript No. JCEIT-23-93819 (R);

Published date: 31 May, 2023, DOI: 10.4172/2324-9307.1000270

### Abstract

Humankind has never experienced the geometric growth and interest in computing particularly in Artificial Intelligence (AI) like in recent times. Efficient processes, automated systems and improved decision making are some of the milestones of this trend. The increased demand for text related solutions from generation, learning, classification and several other tasks has motivated the use of different techniques and tools of AI. Creative text ideas have been sought after for innovation, problem solving and improvements. These ideas are needed in all endeavours of life as coming up with them can be a daunting task for individuals, organizations and ideation platforms. In this work, an idea generation system based on improvements of Markov chains approach using a corpus of text is presented. First, a web system was created to collect solutions from people on a case study problem. They were required to make submissions based on purpose and mechanism with examples to guide them. Next, the collected text was clustered based on similarity measure into groups, then abstractive summaries of the respective groups were computed. Markov chains model was then used to generate new text from the submitted text corpus and the most similar Markov chains generated text was compared with each clustered group's abstractive summary using a similarity measure and returned as an idea result. Finally, a pipeline to execute all the components of the system at once was developed. The result was sent for human evaluation based on the metrics of quality, novelty and variety and compared with output from a generative transformer system using the same text corpus and this work's system performed better. The work majorly addresses a challenge faced by ideation platforms.

**Keywords:** Markov chains; Text generation; Ideation; Computational creativity; Artificial intelligence

### Introduction

There is no generally acceptable definition of ideas, however, in most domains, ideas require exchange and analysis for open innovation. While an innovation is more than an idea, an idea has to be adopted to be fully considered an innovation [1]. Creativity is the ability to come up with ideas or artefacts that are new, surprising and valuable. These ideas can be general and can also be domain specific. Ideas are present in every aspect of life which emanate from

psychological or historical creativity [2]. Computational creativity analyses what can be considered creative if carried out by a human. The goal of which is to simulate human creativeness using computational approaches. Since the inception of computing, machines have been programmed to do only what they are required to do but with computational creativity, some level of adaptation can be infused into such systems for a creative result [3]. Artificial Intelligence (AI) over time has been addressing the needs and challenges in the society. The AI systems have demonstrated capability to learn and solve problems earlier exclusive to humans. AI methods such as Recurrent Neural Networks (RNN), long short term memory, etc. have been used for diverse text generation tasks [4-7]. Despite the advances made in text generation, much still needs to be done in text ideation. Creativity has just been fully accepted as a sub-field of AI after long years of neglect. Even when it was embraced, the progress has been slow except in recent times. There is also a challenge of coming up with novel and useful ideas. Most ideation platforms for the purpose of generating ideas often resort to getting input as ideas in form of contest or community contribution approach which is expensive. These ideas or solutions to a given problem are usually submitted through portals popularly referred to as idea portals [8]. Our system generates random ideas in an automatic fashion from an existing corpus or past submissions. Since not much has been done in the approach of automatically generating creative text from a corpus of text using machine learning approaches, the few available related work are characterized by limitations which motivate improvements [9,10].

From the foregoing, there is a need for automatic creative text generation. This motivated the design and implementation of an idea generation system for this work especially for addressing the challenge of ideation platforms. The system uses a clustering by similarity measure of a corpus of text. The clusters are in groups. This corpus is gotten from solutions to a case study problem collected from a separate web system. Abstractive summarization of each cluster group is computed. Then Markov chains automatically generate new text from the text corpus. The similarity measure compares the abstractive summary of each group with the Markov chains generated text. The output with the highest similarity score is returned as a new idea. A pipeline executes these sub-processes at once. The new ideas can be used for different purposes and systems. The main contributions of this work are:

- Abstract and concept representation of clusters of text.
- Selecting best Markov chains generated text.
- Automatic idea generation using a pipelined system.
- Providing an efficient and economically viable system for crowd ideation platforms.

The rest of the paper is organised as follows: Section 2 presents related work about similar approaches to generating ideas based on machine learning. Section 3 provides an in-depth description of our work with the implementation. Human evaluation of the work is presented in section 4. Section 5 provides a conclusion of our work with proposals for future directions.

### Related work

There is an increasing demand for text related solutions over the years. The text generation aspect of artificial intelligence has provided models for creating tokens which are discrete in nature. These models

have various implementations which are of diverse practical use. Different frameworks such as recurrent neural network, long short term memory, natural language generation and several others have been used to generate text. Over time, ideas have inspired new ways of doing things which include innovative text generation [11]. Describe ideas as a part of any innovation which involves generation and selection. Every domain has a peculiar method to this process. Identifying the appropriate opportunity is a determining factor to how good an idea generation process will turn out. Idea generation process tends to be a problem solving process which has a goal to attain high quality in the output based on the peculiarity of the demand or requirements used machine learning in creating recipes. He generated quantities of ingredients automatically and possible preparations with the help of a machine learning technique with experts assessing the flavour and determining the structure for combinations. He used a data set of recipes from the web and ingredients which are extracted, generator for generating random combinations and genetic algorithm for transformations. He also used a selector which based on a fitness function selects the best generators. The preparation steps are derived from the generating and selecting from the combinations. The quantification is used to select the right quantities of ingredients to the recipes. The final result is a recipe with the right amount of ingredients and a preparation procedure. The creativity of the recipes is in how ingredients are combined as well as the cooking techniques. This approach has limitations such as requiring experts to assess the flavour and determining the structure for combinations. This introduces bias as what is best for the experts may not be universally accepted. The approach also requires a long time of computation [12]. In his thesis studied idea generation based on machine learning and natural language processing. He started first by conceptualizing ideas as problem-solution pairs, then computation of novel ideas from articles published in journal of science. The dataset is splitted and used for idea detection using neural networks, argumentative zoning, citation sentiment analysis, and idea flow analysis. The part which involve idea generation requires noun-phrase extraction using part-of-speech tagging algorithm, classification of the extracted noun phrases into problems and solution pairs, aggregation of high co-occurrences, construction of ground truth and carrying out collaborative filtering by predicting new links between pairs that do not occur together in the same abstract. Again, this system has limitations such as using dataset with human labeled ground truth, since tagging every text is not possible, tagging abstract of the scientific papers is done as this limits the comprehensiveness of the result, the constraint on pairing and search for missing links between abstracts, filtering of training data, tagging, and large corpus requirement are worthy of note. There is also a lack of interface for interactive computation.

### Idea generation

In this section, the approach used towards the realization of the idea generation system is presented.

## Materials and Methods

### Overview of the system approach

To generate new ideas, a corpus of data is required. Clustering the text based on similarity metric is followed. Computation of abstractive summarization of the clusters is the next step. Then Markov chains text generation and finally selecting the final result from similarity comparison between the Markov chains generated text and the

abstractive summary. In Figure 1, the architectural flow of the system is presented. The next subsections present a description of the design components and background of the system.

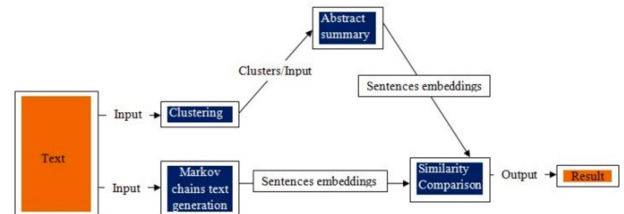


Figure 1: A diagram showing the architectural flow of the system.

**Ideas collection:** The ideas are collected from a separate web system and used in the system. The architecture of the system is the typical three layer architecture. This system is not integrated to the ML system, only the data output is used. The entries are based on possible solutions to a case study problem. In other to guide submissions, users are required to split their solutions into purpose and mechanism. Purpose describes what an idea entails and mechanism states how the idea can be applied.

**Clustering:** The submissions are clustered into groups as each group comprises of similar sentences universal sentence encoder is used for determining similarity by using pretrained sentence encoding model [13]. Sentence embedding emanate from word embedding which refer to the representation of knowledge where a word is represented by a vector. Vector here refers to arrays of numbers. Similar words have similar embeddings as the embedding represents floating point values as a dense vector and can be of any dimension. This has provided the basis for which datasets in texts can be learnt. Word embeddings show which words are geometrically closer to one another as their distance can be seen on the observed space and also their respective relationships. Following the breakthrough in word embeddings, a sentence embedding model was proposed. This model converts text into fixed length vector representations as the model is called a universal sentence encoder. The universal sentence encoder uses tensor flow to encode text into higher dimensional vectors. The encoding contains useful semantic information. After the vectors of each sentence is computed, the vectors are used to determine the cosine similarity which is calculated as the angle between two vectors. For smaller values of two vectors, the vectors are reported to being similar. The angle is computed from taking the arcos of the cosine similarity. Hence similarity is given as:

$$S(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \tag{1}$$

Considering vector embeddings of sentences A and B, their similarity can be computed from the formula in equation 1. With universal sentence encoder discussed above, an example of the vector embeddings derivation of sentences is shown in Figure 2.

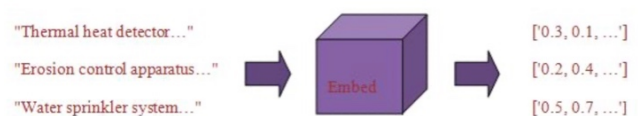


Figure 2: Sentence embedding depiction adapted from.

**Abstractive summarization:** Based on the clustered groups, a summarization procedure is used, particularly abstractive summarization. There are two types of summarization techniques namely; abstractive and extractive summarizations. Abstractive summarization has provided more interesting results over the years than the extractive summarization. Extractive summarization involves getting important information which could be a sentence from a given text or data. In selecting useful sentences, approaches in statistics and/or linguistics are applied. Abstractive summarization as a process finds new concepts and generates new output with fewer words [14]. Classical approaches and neural methods have changed the way abstractive summarization is conducted. Classical approaches comprise content selection, surface realization and information extraction as pipe lined tasks. Neural methods typically learn from a source document and generate a networked-summary. Encoder-decoder framework, attention mechanism, and reinforcement learning are some of the neural approaches for abstractive summarization [15]. Advances in automatic abstractive summarization received a boost with Google’s text to text transfer transformer. The transformer serves as a pretrained data source eliminating the process of training and testing a model with specific data. This is achieved with the help of a corpus satisfying all requirements of massiveness, diverseness, and highest quality. This pretrained dataset is then used for the text to text model of the abstraction. Using this method has proven to save time and computational cost [16].

**Markov chains text generation:** Text generation using Markov chains is a chance process which involves transitioning from one state to another without taking into consideration the past states [17]. Following from the Markov modelling process entails representing temporal patterns which are determined by chance probabilistic outcome over sample observations. The process moves from one state to another after beginning from one of the states. The set of steps are represented by  $S = \{s_1, s_2, \dots, s_n\}$ . From step 1 to step 2, the chain of movement is accomplished with a probability denoted by  $P_{ij}$ . As shown in equation 3, this probability determines the other next steps of movement. Every move is called a step. An initial probability distribution defined on step S specifies the starting state. The transition probability of moving to state  $S_j$  provided the previous state is  $S_i$  is represented as follows;

$$\Pr(i_j) = \Pr(S_j|S_i) \tag{3}$$

That is, the information that is gotten from a recent past is more useful than a distance past. In equation 3, the conditional probability of being in is shown as a transition which involves moving to state j provided the previous state is i.

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{bmatrix} \tag{4}$$

The transition matrix P formed from transition probability  $P_{ij}$  shown in 4 has non-negative elements such that;

$$\sum_j P_{ij} = 1$$

That is, the sum of elements on each row of the matrix yields 1.

The Markov chains text generation process used in this work involves creating a dictionary with a unique word as key and the following word as value appended in an order determined by probability. The sequence of observations or occurrences and pattern of outcome is subject to chance [18].

### Implementation

**Data collection:** 200 responses to a case study problem were collected separately from a web system designed with python flask framework and hosted on the free Heroku hosting service. The database used is SQL Alchemy. The web system is available online and can be accessed on [ideasubmission.herokuapp.com](https://ideasubmission.herokuapp.com). Submissions were categorised into purpose and mechanism. Participants from diverse background submitted their solution(s) to this question:

We are searching for an innovative (technical) solution for the security of city buildings. In the first step think of possible dangerous events, which may occur (e.g. fire, flood, earthquake, etc.). Then brainstorm innovative solutions, how people in the building could be protected from such danger or rescued from the building.

On successful submission, the data is sent to the database where Postico linked with Heroku is used to read the data. The collected text corpus is then used in the system.

### Clustering

Universal sentence encoder is used to group sentences using measure of similarity scores. The universal sentence encoder package used is from tensorflow.js. After the sentence embeddings are computed, the cosine similarity of each vector result is calculated and compared with the set threshold. The result is groups of similar sentences. The fine tuning is done by choosing a value for the desired threshold at the front end of the system. The 200 sentence input text collected from the web system is clustered into groups automatically based on their similarities. Sentences with similar similarity scores are grouped together.

### Abstractive summarization

A transformer library is used for this sub-task. Python torch is used for the implementation. The pretrained transformer t5 data source is used with 't5-small' pretrained model. With the help of this model, training, testing and validating input data from scratch is not required. The summarized result is displayed after fine tuning is done.

### Markov chains text generation

The 200 sentence text from the separate web system is used as data source for the text generation. The words of each sentence form a pair that is used for the state transitions Markov chains are known for. If after breaking the sentences of the corpus and say a particular first

word is followed by a second word then there is a probabilistic state transition from the first word to the second word. This transition is what had been earlier referred to as Markov chain. The transition probabilities are determined by selecting distinct words in the corpus and assigning keys to them which forms a dictionary. Words that come after any key refers to the value of such a key. In this way, the chain will be iterating through the words, adding any word to the dictionary if not there and adding next word as transition words. The starting word is randomly chosen from the corpus and the desired length of the output. The sampling and iteration is done until the length is reached for the new generated text.

### Similarity comparison

Sentence transformer is used for the comparison. The pretrained model 'bert-base-nli-mean-tokens' is also used. The toolkit 'NLTK' is used for tokenization and other NLP tasks. The cosine similarity computes similarity measure from the vectors of the sentences embeddings. Each of the markov chains generated sentences is compared with the abstractive summary of each group and the sentence with the highest similarity score is returned as the final result for each group.

### The result output

Pipeline executes all the components outlined above at once. Google colab development environment is used for running and deploying the system at development level with tools and libraries described in section 0.3. Figure 3 shows an example of an output from our system.

```

Group 2
Idea: evaporator for removing smoke, heat and combustion gases from a burning structure
concrete fire-resistant floor and air-mechanic compartments area horizontally
Fire Stopping sealant concealed leaving holes in a duct that will provide another source of fire spread in a building
Special fire-resistant ceiling that protect upper structure from the space below
Fire rated glazing systems for lightness and airiness
Photoelectric smoke detector for alerts and warnings
Air aspirating smoke detector for alerts and warnings
Thermocouple sensor heat detector for identifying high temperatures
Automatic sprinkler system that detects the fire's heat, initiates alarm, and begin suppression within moments after flames appear
Water mains for conveying water through the passages for firefighting operations
Automatic water control to serve as secondary source water for firefighting operation
Inert gases used to extinguish fire in confined areas or enclosure with the structural
dry chemical system to extinguish fire in confined industrial processes, equipment, machinery and paint spray booth, area
Hepatic air fire prevention system
New Reel used to disrupt fire chain
Fire hydrant used by firefighters to tap water into the water supply when extinguishing fire
Flarelighting lights to light access to the building and facilities for fire fighters Dry riser and fire point signs to give information on fire equipment
.....
Abstract Summary: fire-resistant well used to remove smoke, heat and combustion gases
Markov Text:
sentence 0: survival surviving wind from a burning structure smoke exhausting air-conditioning system to prevent a storm surge or spring tide from flood
erosion control
sentence 1: create mechanisms in the case of fire flood resilience waterproofing and shock prevention avalanche control blizzard control flood freezing
control tornado damage control
sentence 2: fire wet chemical system to prevent water from passing beyond the defined area for flood protection levee system to prevent water from p
operation of the structure
sentence 4: repairing the beach barrier for containing floods erosion control apparatus for operating during earthquake use cellular during a landslide us
knowbest protection against fumes
sentence 5: communication device for objects apparatus for detecting failure of a float switch foam fire suppression for extinguishing inert gases used to
rotary foam distributor for providing
sentence 6: have superabsorbent food handy take refuge during a landslide use a fire dry chemical extinguisher designed to combat fires in kitchen cooki
excluding dangerous area
sentence 7: conventional method implementation of a certain magnitude create mechanisms in the masonry that are activated when detecting a p wave elast
barriers on
sentence 8: horizontally fire stopping sealant concealed leaving holes in a building gypsum fire-resisting ceilings that protect upper structure from th
for exterior weathering e-collings
sentence 9: strike exposure looking out for the elderly and children using robots during rescue use of lowcost seismic isolators improve the confinement
destructive capacity sensor
sentence 10: dry chemical system to extinguish fire involving flammable substance rotary foam distributor for providing information to the rack without
sentence 11: silent during earthquakes for detecting initial shock and activating preliminary operation information processing apparatus to monitor an o
to detect fires notify occupants summon
sentence 12: from an area to be used for class b and c fire wet chemical system to draw harmful smoke from rooms in an airconditioned building by way
sentence 13: preventing beach erosion directing water in flood areas and providing highway barriers for guidance and safety reportback system for persi
camera for facilitating building security
sentence 14: from disasters post traumaticheadache management contaminated and medical waste management
sentence 15: objects apparatus for detecting earthquake using accelerometer earthquake warning hurricane obstructions check and update hazardous contan
mitigating earthquake effect use of carbon fibre
sentence 16: wet chemical system to extinguish fire in confined areas or enclosure with the structural integrity to retain extra water terracing and th
storing water
Idea Result: wet chemical system to extinguish fire in confined areas or enclosure with the structural integrity to retain extra water terracing and t
for storing water with score :0.83387
    
```

Figure 3: Sample output of ideas generated from our system.

System	Distribution	Percentage
System 1	5	25%
System 2	15	75%

Table 1: Evaluation summary of the system in comparison to GPT-2 system based on quality evaluation.

Figure 4 shows the statistical distribution of the respondents for both systems based on quality evaluation.

### Evaluation

Three different human evaluation metrics of the system are presented together with the procedure, result and discussion. The evaluation metrics of quality, novelty and variety were used.

**Procedure:** Evaluation is carried out by 20 people from different disciplines. They were carefully chosen based on their backgrounds [19]. They compared the result of the idea generation system with result from OpenAI's Generative Pre-trained Transformer 2 (GPT-2) system using the same text corpus. The evaluation was done based on quality, novelty and variety [20].

### Quality

As pointed out by quality as assessed for each output of both systems measures the extent to which an output makes semantic sense. In this evaluation, the participants were shown results from the two systems without a description of what they are to avoid bias. The verdict on which is best is collected. A short description explaining what makes a quality text was included. This served as a guide for those completely unfamiliar with this metric.

### Results and Discussion

In the quality evaluation exercise, the output of the system is compared with result from OpenAI's Generative Pre-trained Transformer 2 (GPT-2) and the result is presented as shown in Table 1. As can be observed, 75% of the respondents saw the idea generation system's output as best in quality as against the 25% for the GPT-2 system.

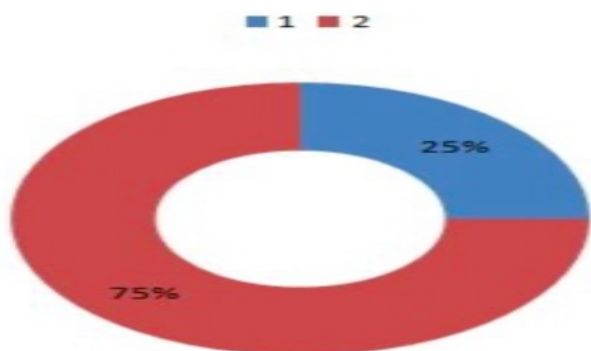


Figure 4: Statistical distribution of quality evaluation of the two systems.

### Novelty

Novelty of an idea describes how uncommon the ideas appear to the respondents. The result from both the idea generation system and the GPT-2 system are compared on this basis and verdict on which is best was collected. Again, respondents were given short description of what a novel idea means.

In this evaluation, the output of the system is compared with output from OpenAI’s Generative Pre-trained Transformer 2 (GPT-2) system and the result is presented in Table 2; as can be observed, 75% of the respondents saw the idea generation system’s output more novel compared to the 25% for the state of the art GPT-2 system.

System	Distribution	Percentage
System 1	5	25%
System 2	15	75%

Table 2: Evaluation summary of the system in comparison to GPT-2 system based on novelty evaluation.

Figure 5 shows the statistical distribution of the respondents for both systems based on novel evaluation.

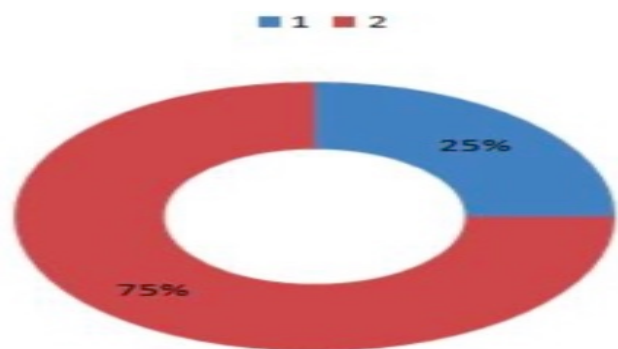


Figure 5: Statistical distribution of novelty evaluation of the two systems.

### Variety

Variety implies dissimilarity in the systems’ output. In the comparison of the systems, respondents were required to check how varied each system’s result is and rate which is best.

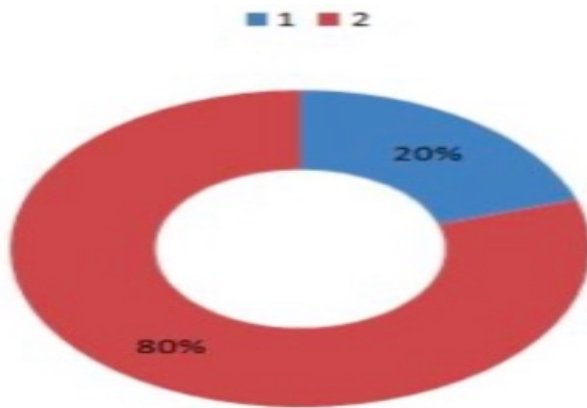
In determining the variety of the thesis’ system, its output is compared with output from OpenAI’s Generative Pre-trained Transformer 2 (GPT-2) system and the result is presented as shown in Table 3.

System	Distribution	Percentage
System 1	4	20%
System 2	16	80%

Table 3: Evaluation summary of the system in comparison to GPT-2 state of the art based on variety evaluation.

As can be observed, 80% of the respondents saw the idea generation system’s output as more varied compared to the 20% for the state of the art GPT-2 system.

Figure 6 shows the statistical distribution of the respondents for both systems based on variety evaluation.



**Figure 6:** Statistical distribution of variety evaluation of the two systems.

## Conclusions

This paper presents an idea generation system based on improvements of the Markov chains text generation approach. A text corpus was gotten from a developed web system that collects solutions from people based on a case study problem. The collected text corpus was then passed to the idea generation pipelined system which does the following:

- Cluster the collected text corpus into groups of similar sentences based on similarity measure.
- Generate abstractive summary of each clustered group.
- Use Markov chains model to generate new text sentences from the text corpus.
- Compare the Markov chains generated sentence text with the abstractive summary of each group and return the most similar sentence text as result.
- Use a pipeline to deploy the work where all the components are executed at once.

## Limitation and Future Work

The result of the pipeline are new ideas. This result was sent for human evaluation and was compared with result from similar generative system using the same text corpus. Based on the evaluation metrics of quality, novelty and variety, our system produced more creative ideas with the highest scores. Since ideas are needed in all walks of life, the system is a useful tool for individuals, ideation platforms, organizations, innovators, etc. Future work should be focused on improving the abstractive summarization for new words for the similarity comparison. Also, evaluation of our system was done using a generative pretrained transformer system.

## References

1. Riedl C, May N, Finzen J, Stathel S, Kaufman V, et al. (2011) An idea ontology for innovation management. *Sem Ser, Inter Web Appli: Emer Conc* 303-321.

2. Pachet F, Roy P (2011) Markov constraints: Steerable generation of Markov sequences. *Constraints* 16:148-172.
3. Colton S, Wiggins GA (2012) Computational creativity: The final frontier?. *InEcai* 12:21-26.
4. Alex Graves (2014) Generating sequences with recurrent neural networks. *arXiv e-prints* 1-10.
5. Pawade D, Sakhapara A, Jain M, Jain N, Gada K (2018) Story scrambler automatic text generation using word level RNN-LSTM. *Int J Inf Tech Comput Sci* 10:44-53.
6. Eck D, Schmidhuber J (2022) A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intell Arti* 103:48.
7. Zhu Y, Lu S, Zheng L, Guo J, Zhang W, et al. (2018) Tegygen: A benchmarking platform for text generation models. *41<sup>st</sup> Int ACM SIGIR Conf Res Develop Inf Retr* 1097-1100.
8. Finzen J, Kintz M, Kaufmann S (2012) Aggregating web based ideation platforms. *Int J Technol Intell Plan* 8:32-46.
9. Xiang L, Yang S, Liu Y, Li Q, Zhu C (2020) Novel linguistic steganography based on character level text generation. *Mathematics* 8:1558.
10. Cerquetti R, Ficcadenti V, Dhesi G, Ausloos M (2022) Markov chain Monte Carlo for generating ranked textual data. *Infor Sci* 610:425-439.
11. Girotra K, Terwiesch C, Ulrich KT (2010) Idea generation and the quality of the best idea. *Manage Sci* 56:591-605.
12. Cer D, Yang Y, Kong SY, Hua N, Limtiaco N, et al. (2018) Universal sentence encoder. *arXiv preprint*.
13. Lladser ME, Betterton MD, Knight R (2008) Multiple pattern matching: A Markov chain approach. *J Math Biol* 56:51-92.
14. Madhuri JN, Kumar RG (2019) Extractive text summarization using sentence ranking. In: *Proceedings of International Conference on Data Science and Communication* 1-3.
15. Lin H, Ng V (2019) Abstractive summarization: A survey of the state of the art. *Proc AAAI Conf Artif Intell* 33:9815-9822.
16. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, et al. (2020) Exploring the limits of transfer learning with a unified text to text transformer. *J Mach Learn Res* 21:5485-5551.
17. Regnier M, Szpankowski W (1998) On pattern frequency occurrences in a Markovian sequence. *Algorithmica* 22:631-649.
18. Guevara C, Bonilla D (2021) Algorithm for preventing the spread of COVID-19 in airports and air routes by applying fuzzy logic and a Markov chain. *Mathematics* 9:3040.
19. Radford A, Wu J, Child R, Luan D, Amodei D, et al. (2019) Language models are unsupervised multitask learners. *OpenAI blog* 1:9.
20. Han J, Shi F, Chen L, Childs PR (2018) The combinator-a computer based tool for creative idea generation based on a simulation approach. *Des Sci* 4:e11.