iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44283

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Text Similarity Using Siamese Networks and Transformers

Bharath Chandra Chikoti[1], Kushwanth Jeelaga[2], Aryan Srivatsava Dande[3], Sunil Kumar[4]
[1,2,3]*Student,* [4]*Professor, Department of Electronics and Computer Engineering*
Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India

Abstract-- In result-oriented conversational models like message renders and chatbots, finding the similarity between the input and output text result is a big task. In general, the conversational model developers lean to provide a minimal number of utterances per instance, and this makes the classification a difficult task. This problem becomes more difficult when the length of the processed text per action is short and length of the user input is long. Identical sentence pair detection reduces manual effort for users with high reputation. Siamese networks have been one of the best innovative architectures designed in the field of natural language processing. A Siamese network was initially designed for computer vision applications. Later the core concept of this algorithm was designed for NLP ,to identify similarity for two given sentences. Siamese networks are used in this algorithm. It's an artificial neural network also known as a twin neural network that works in tandem on two independent input vectors to calculate equivalent output vectors using the same weights. Also there are few commonly addressed drawbacks like word sense disambiguation and memory intolerance of initial inputs for sentences having more than 15-20 words. To tackle these issues, we propose a modified algorithm that integrates the transformer model implicitly with the core part of the siamese network. Transformer model helps to generate each output position based on the semantic analysis of overall sentence and can also deal with homonyms, by extracting its meaning based, which is syntactic based and semantic based on the overall sentence or paragraph or text.

Keywords: **Message Renders and Chatbots, Siamese Networks, Natural Language Processing, Transformer Model, Artificial Neural Networks.**

I. INTRODUCTION

Natural Language Processing-NLP, is computational linguistics branch of AI, is such a technology which is gaining the interest of many developers, scientific researchers as it got perfect blend of machine learning, language, and also artificial intelligence. Jaccard similarity and Cosine similarity are two methods for calculating syntactic similarity in text. With the enormous amount of textual information generated every day, NLP has piqued the interest of many developers and scientific researchers in the hope of developing methods to process this information in order to make it more efficient, accessible, and comprehensive. To describe by an example, in the domains, like newspaper articles, intellectual property, that involves a lot of textual data and paperwork, it is the task of NLP which will cluster the similar textual data and papers, which simplifies document analyses for the users. This kind service from NLP will increase productivity by reducing the processing time, and also this will also improve the precision with which large patent systems are handled and analyzed. Any document is generally divided into introduction, context, references, and description of pics or graphs. Usually the context and description are considered as the important part of a document, the introduction will summarize the overall technology described in the document. Similarity analysis of a document may include any of the structural parts including introduction, context, references, and others. But, in many cases, it is desirable that the documents with similar context can be clustered together for the analysis.

To find a solution to this problem, one must need insight to define the concept of similarity in a quantitative manner. In general, semantic similarity and syntactic similarity are two major similarity metrics encountered in text similarity analysis. Semantic similarity, on the other hand, is concerned with the interpretation-based similarity and meaning coherence of the two given texts. The idea behind syntactic similarity is that the similarity between two texts is proportional to the total number of identical words in them; however, precautions must be taken to ensure that the chosen method does not become biased towards the given text with a higher word count. While the syntactic similarity value can be calculated by constructing measures based on the word counts of the two inputs, the semantic analysis employs a more advanced method that employs word representations to extract meaning-based values for the two texts.. Before continuing with the analysis, the texts must be pre-processed to remove all unwanted tags, interpretation-based or other predefined characters or animated words. The pre-processed input text is thus reduced to the respective word roots, which are words, and lemmatization is performed on the text to be analyzed.

A significant part of NLP highly relies on the connection in high dimensional vectors. Generally an NLP processing will take any textual data, prepare it to generate a enormous vector or array rendering that text and then make required transformations.

II. RELATED WORK

For cQA, a Siamese Convolutional Neural Network was used. Deep convolutional neural networks are used as twin networks in the SCQA architecture, with a contrastive energy function at the top. Parameter sharing ensures that in the semantic space, the question and any relevant response are closer together, while the query and any irrelevant answer are far apart. For example, "President of the United States" and "Barack Obama" should be closer than "President of the United States" and "Tom Cruise lives in the United States." SCQA, which is frequently difficult to obtain in big quantities, requires similar question pairings. SCQA is made up of two deep convolutional neural networks (CNN) with convolution, max pooling, and rectified linear (ReLU) layers, as well as a fully connected layer at the top. CNN generates a nonlinear projection of the question and answer term vectors in semantic space. The semantic vectors that result are linked to a layer that computes their distance or similarity. The contrastive loss function combines the distance measure and the label. The gradient of the loss function is calculated using back-propagation with respect to the weights and biases shared by the sub-networks. The stochastic gradient descent method is used to update the parameters of the sub-networks.

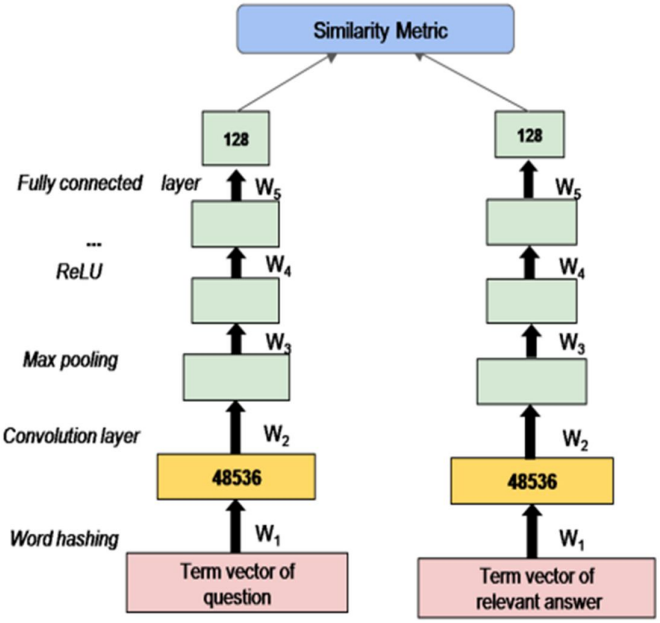


Figure 1: Siamese network in CNN mode

The task of calculating the similarity between two texts utilizing both direct and indirect relationships is known as measuring semantic textual similarity. Siamese architectures are good for these jobs because it makes logical to use a similar model to handle comparable inputs when the inputs are similar. The networks will then have representation vectors with the same meanings, making it easy to compare phrase pairings. Because the weights are shared across sub networks, there are fewer parameters to train, resulting in less training data and a lower risk of overfitting. Given the amount of human-labour necessary to create STS datasets, Siamese neural networks may be the best option for the STS challenge.

III. PROPOSED SYSTEM

Information retrieval, text categorization, file clustering, topic detection, subject matter tracking, questions technology, query responding, essay scoring, rapid answer scoring, gadget translation, textual content summary, and others all use textual-content similarity measures. Finding word similarity is an important element of text similarity, which is subsequently utilized as a starting point for sentence, paragraph, and record similarity. In terms of lexical and semantic techniques, phrases can be comparable. If two words share the same character sequence, they are lexically related. Words are semantically comparable if they have the same component, have different meanings or semantics, are employed in the same way, and are used in the same context.

Multi-head interest is an attention mechanism module that runs through the mode of an attention mechanism many times in parallel. The independent interest outputs are then stacked horizontally or vertically, and the anticipated dimension is linearly converted. Multiple attention heads, intuitively, allow access to elements of the collection in a different way (e.g. longer-time period dependencies versus shorter-term dependencies).

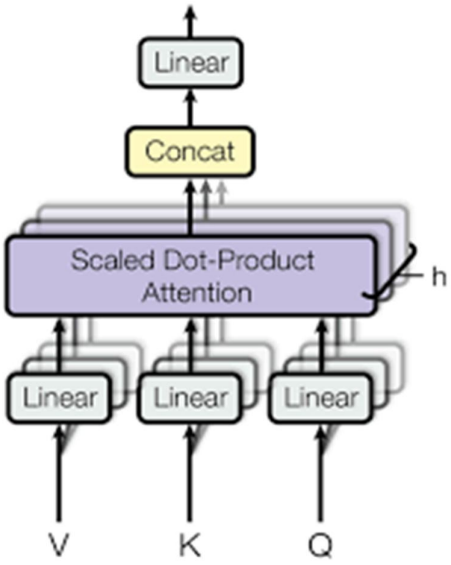


Figure 2: Multi head attention

IV. IMPLEMENTATION

A. Data Set Collection

The target of this undertaking part is to find out which of the feasible pairs of input sentences contain the inputs with the same which means. The labeling is the set of labels which have been manually detailed. The main truth labels are inherently detailed, as the genuine which means of inputs can by no means be identified with actuality. Manual labeling is likewise a 'noisy' manner, also affordable human beings will never accept. Hence end result, the ground reality outputs in the data is be considered to be 'knowledgeable' however not a hundred% precise, and can encompass false output labeling. We accept as true with the output labels, the whole part, to symbolize a meaningful consensus, but this will no longer be real in a case by means of case basis for man or woman gadgets with in the dataset. Textual content summarizing is a design to condense the big quantity of facts right into a reasonable shape by means of the technique of selection of critical data and removing irrelevant and redundant data. With this quantity of human language interpretable records which are inside the international wide area of textual content summarizable process is emerging to be very important.

```
N=len(data)
print('Number of question pairs: ', N)
data.head(10)
```

```
Number of question pairs:  404290
```

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and c...	I'm a triple Capricorn (Sun, Moon and ascendan...	1
6	6	13	14	Should I buy tiago?	What keeps childern active and far from phone ...	0
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
8	8	17	18	When do you use シ instead of し?	When do you use "&" instead of "and"?	0
9	9	19	20	Motorola (company): Can I hack my Charter Moto...	How do I hack Motorola DCX3400 for free internet?	0

Figure 3: Dataset representation

B. Importing NLTK and Tokenizing Sentences

NLTK is a word tokenizer library which also has many other functions built in it. First one chooses the best question pairs that are duplicate to train the version of the original one. We construct two batches as input for the Siamese community and we count on that query q1i (query i inside the first batch) is a duplicate of q2i (question i inside the second batch), however all other questions in the second batch aren't duplicates in qli. The check set makes use of the authentic pairs of questions and the fame describing if the questions are duplicates.

C. Converting Tokens into a Tensor

This technique may be carried out in three steps ;

- Word mapping of textual content (token) to an index
- Creation of unique tokens for stop-of-sentence, padding and out-of-vocabulary symbols
- Conversion of phrases in the dictionary into tensors

The vocabulary will organize our facts in key: value pairs where the important thing will be the time period and the key indices can be an integer index related to that time period. There may be unique tokens so one can have those traits

__PAD__: suggests the padding symbol. This work can be very useful because it will provide a template for the creation of the input for our records generator in an smooth and efficient way.

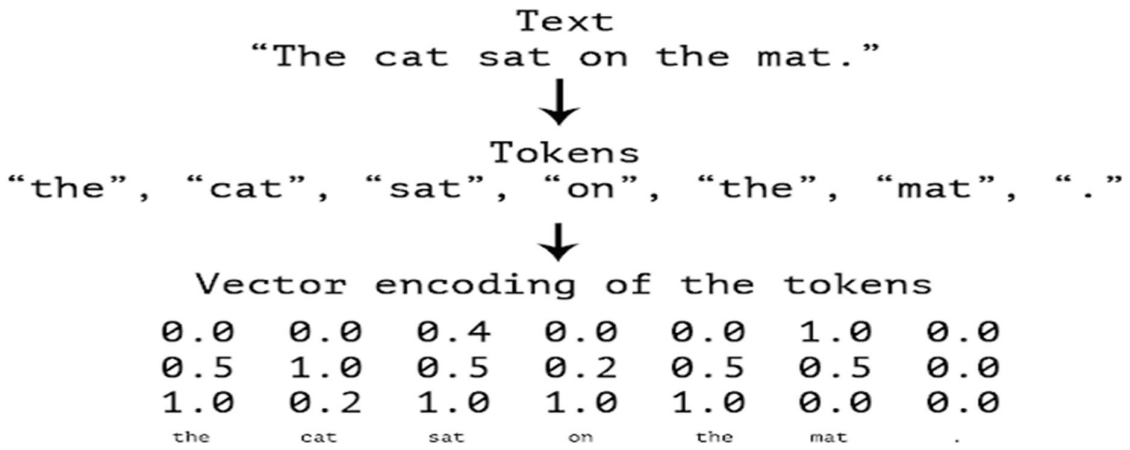


Figure 4: Word Tokenization

D. Designing a Data Generator

Maximum of the time in Natural Language Processing, and AI in standard we use batches while education our statistics units. If stochastic gradient descent is used with one example at a time, it's going to take forever to build a model. In this case, it suggests tips on how to construct a information generator that takes in Q1 and Q2 and returns a batch of length batch_size inside the following format ([q11,q12,q13,...], [q21,q22,q23,...]). The tuple includes two arrays and each array has batch_size questions. again, q1i and q2i are duplicates, but they are no longer duplicates with every other factors within the batch.

The command next (data_generator) returns the next batch. This iterator returns the statistics in a layout that you may at once use for your model whilst computing the feed-forward of your algorithm. This iterator returns a couple of arrays of questions.

The generator must return shuffled batches of information of inputs from databases or memory. To acquire this without editing the real query lists, a list containing the indexes of the questions is created. This list can be shuffled and used to get random batches every time and whenever the index is reset.

E. Designing a Normalization Function

$$\hat{X}[j,:] = \frac{X[j,:]}{||X[j,:]||}$$

Figure 5: Normalization formulae

One needs to perform Scaling to unit duration shrinks/direction a vector (a column of facts can be considered as a N-dimensional vector form) to a single tensor. It is used on the entire data set, the converted statistics can be represented as a gaggle of one dimensional vectors with specific instructions towards the N-dimensional unit sphere.

Now, Generalizing/normalizing is certainly a large time period and every one of them will have execs and cons! I'll most effective attention on improvisation in this text in any other case this article will go manner too long.

F. Designing a Siamese Model

A Siamese model is a network designed which uses the weights as same as before on the equal time as running in parallel on two precise entry vectors to process same dimensional output vectors.

The question embedding is produced, run it via an LSTM layer, normalize v1 and v2, and ultimately use a triplet loss (defined under) to get the corresponding cosine similarity for each pair of questions. As conventional, it's miles started out by using importing the statistics set. The triplet loss uses a base (anchor) input vector that is now compared to a powerful (ground) entry and a irrelevant input. Gap due to the base (anchor) input vector to the extremely similar (ground) input is minimized, and the gap due to the base(anchor) input to the terrible enter is maximized. In math equations, the under proper hand expression must be maximized.

$$L(\text{Anchor},\text{Positive},\text{Negative})=\text{maxium}(\|f(\text{Anchor})-f(\text{Positive})\|2-\|f(\text{Anchor})-f(\text{Negative})\|2+\alpha(\text{alpha}),\text{zero})$$

Here A is the anchor input, as an example q11, P the reproduction input, for example, q21, and N the bad input (the non-reproduction question), as an instance q22. α is a margin; you could reflect on consideration on it as a safety net, or with the aid of how masses you need to push the duplicates from the non-duplicates

G. Creating a Loss Function Using Mean , Closest Factors

One term makes use of the mean of all of the non-duplicates, the second makes use of the closest terrible. Our loss expression is then:

$\text{Loss1}(A,P,N)=\max(-\cos(A,P)+\text{mean\_neg}+\alpha,0)$

$\text{Loss2}(A,P,N)=\max(-\cos(A,P)+\text{closestneg}+\alpha,0)$(three)$\text{Loss}(A,P,N)=\text{imply}(\text{Loss1}+\text{Loss2})$

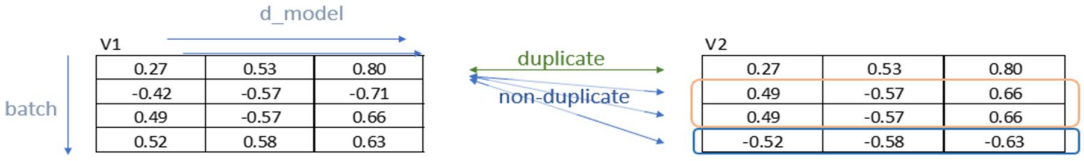


Figure6*:* Similarity matching-1

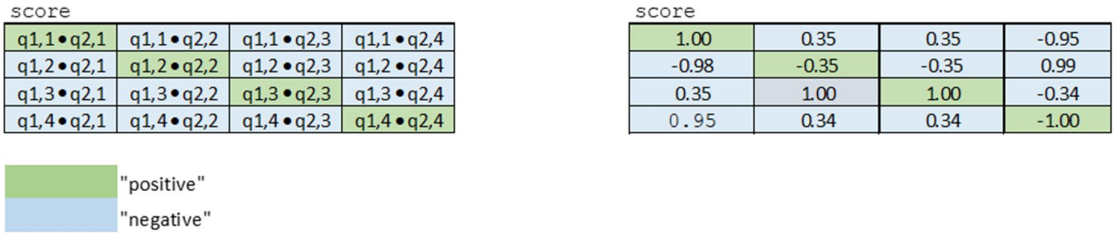


Figure7*:* Similarity matching-2

H. Training the Model and Loading the Parameters

Feature that takes for your model and trains it. To train your version you have to decide how usually you need to iterate over the entire records set; every generation is defined as an epoch. For every epoch, you need to pass over all the statistics, the usage of your training iterator.

- labeled_data=generator
- metrics=[tripletloss()],
- loss_layer=tripletloss()
- optimizer=trax.optimizers.adam (with mastering fee of 0.01)
- lr_schedule=lr_schedule,
- output_dir=output_dirGoogletrax standard training and eval task functions :

I. Importing the Trained Parameters of the Model

Import the parameters from the model that is built in the earlier stage.

```
model1 = Siamese()
model1.init_from_file('/content/model.pkl.gz')
```

```
(((array([[-0.7434783 , -0.5323    ,  0.26556844, ...,  0.24734499,
           0.971258  , -0.3176002 ],
          [-1.9103315 , -1.2298064 ,  0.7929189 , ..., -1.3576206 ,
           -0.9268899 ,  0.11710498],
          [ 1.1356051 ,  1.2533569 ,  1.4670613 , ...,  1.2557949 ,
            1.1703947 ,  1.7554839 ],
          ...,
          [-0.49271938,  0.06522572, -0.74080336, ...,  1.4723355 ,
            1.1603701 ,  0.51038134],
          [ 1.4607905 , -0.15703319,  0.5001072 , ...,  0.18419997,
           -0.5392223 , -0.4307455 ],
          [-1.4316021 , -1.2368174 ,  0.0611912 , ..., -0.24021208,
            0.34730613, -0.07061554]], dtype=float32),
   (((), ((), ()))),
    ((array([[-0.02550941, -0.06643244, -0.03194237, ..., -0.01651929,
              -0.0235158 , -0.02485679],
             [-0.0683428 , -0.06671927,  0.0349182 , ...,  0.01044205,
               0.02273431,  0.0717328 ],
             [-0.03171944,  0.01028203,  0.05781721, ..., -0.04236581,
              -0.04657996, -0.06436575],
             ...,
             [ 0.03127091, -0.02159434, -0.08765983, ...,  0.02809162,
```

I

Figure9: Trained parameters

J. Importing the Text Summarizer

Textual content filtering is a process to filter out the huge content of facts into a precise form by way of the system of finding and choosing of crucial statistics and removing useless and repetitive information facts. Along the quantity of this language data which is inside the global internet the region of textual content summarization is turning into very crucial. The excerpt summarization is the only in which the same sentences present inside the report are used as results.

The excerpt summarization is easier and is the overall exercise most of the automated text filtration experts at the existing time. Excerpt filtration method entails giving rankings to inputs the use of a few approach after which the use of the sentences that gain top scores as filtered short length outputs. As the exact sentence present inside the file is used the text component can be absented which ends up in technology of low observation in depth filtration method. That form is typically unsupervised also language independent too. Also the fact that this sort of summarized part does its work in transmitting the crucial sequence it could not be always easy or fluent. On this occasion there can be no connection among contiguous sentences inside the accurate resulting in the copy lost in readability.

K. Generating the Summarized Sentences

T5 is a brand new transformer version from Google this is educated in an give up-to-stop way with textual content as input and changed textual content as output. It achieves some effects on a couple of NLP responsibilities like summarization, question answering, device translation, more the usage of a text-to-textual content transformer skilled on a massive text corpus. Transformers is used as a version from libraries to summarize any given text. T5 is an abbreviative summarization algorithm. It manner that it's going to rewrite sentences whilst important than just choosing up sentences immediately from the authentic text.

The US has "passed the peak" on new coronavirus cases, President Donald Trump said and predicted that some states would reopen this month.The US has over 637,000 confirmed Covid 19 cases and over 30,826 deaths, the highest for any country in the world.At the daily White House coronavirus briefing on Wednesday, Trump said new guidelines to reopen the country would be announced on Thursday after he speaks to governors."We'll be the comeback kids, all of us," he said. "We want to get our country back."The Trump administration has previously fixed May 1 as a possible date to reopen the world's largest economy, but the president said some states may be able to return to normalcy earlier than that.

Summary from T5:

The us has over 637,000 confirmed Covid-19 cases and over 30,826 deaths. President Donald Trump predicts some states will reopen the country in april, he said. "we'll be the comeback kids, all of us," the president says.

Figure12: Summarization example

V. RESULTS

```
question1 = test_string1
question2 = test_string2

predict(question1 , question2, 0.7, model1, vocab, verbose = True)
```

```
Q1  =  [[ 9030 13394  1870    28  1872  5221   267     6  1708  6584    28  9521
   9522   148  2366   960  4068    55   111   131    78  1857  2059    28
    750   148  2366  3398   131   619     0  1300   622    72  4159  3622
      6  1019    39     0  7287   148  5799   218  1496  2816     0    39
  13394  6870   495    39    78  2061   254  3398   148     1     1     1
      1     1     1     1]]
Q2  =  [[ 9030 13394  1870    28  1872  5221   267     6  1708  6584    28  9521
   9522   148  2366   960  4068    55   111   131    78  1857  2059    28
    750   148  2366  3398   131   619     0  1300   622    72  4159  3622
      6  1019    39     0  7287   148  5799   218  1496  2816     0    39
  13394  6870   495    39    78  2061   254  3398   148     1     1     1
      1     1     1     1]]
d   =  1.0
res =  True
True
```

Figure 13: Output

The paragraphs will be summarized using the imported text summarizer and then after purely preprocessing the summarized output sentences , Siamese model will be given these sentences and input and similarity will be obtained in a scale of 0-1 . The more the result is closer to 1,the more the similarity

VI. CONCLUSION

However there are algorithms and other software applications that determine the similarity between two texts but this algorithm makes use of Siamese Networks and transformer summarizer together to deal with lengthy paragraphs and also this can be fatherly extended and integrated with many other projects like chat auto-prototype answer generation and optimization for named entity recognition as mentioned earlier

VII. ACKNOWLEDGEMENT

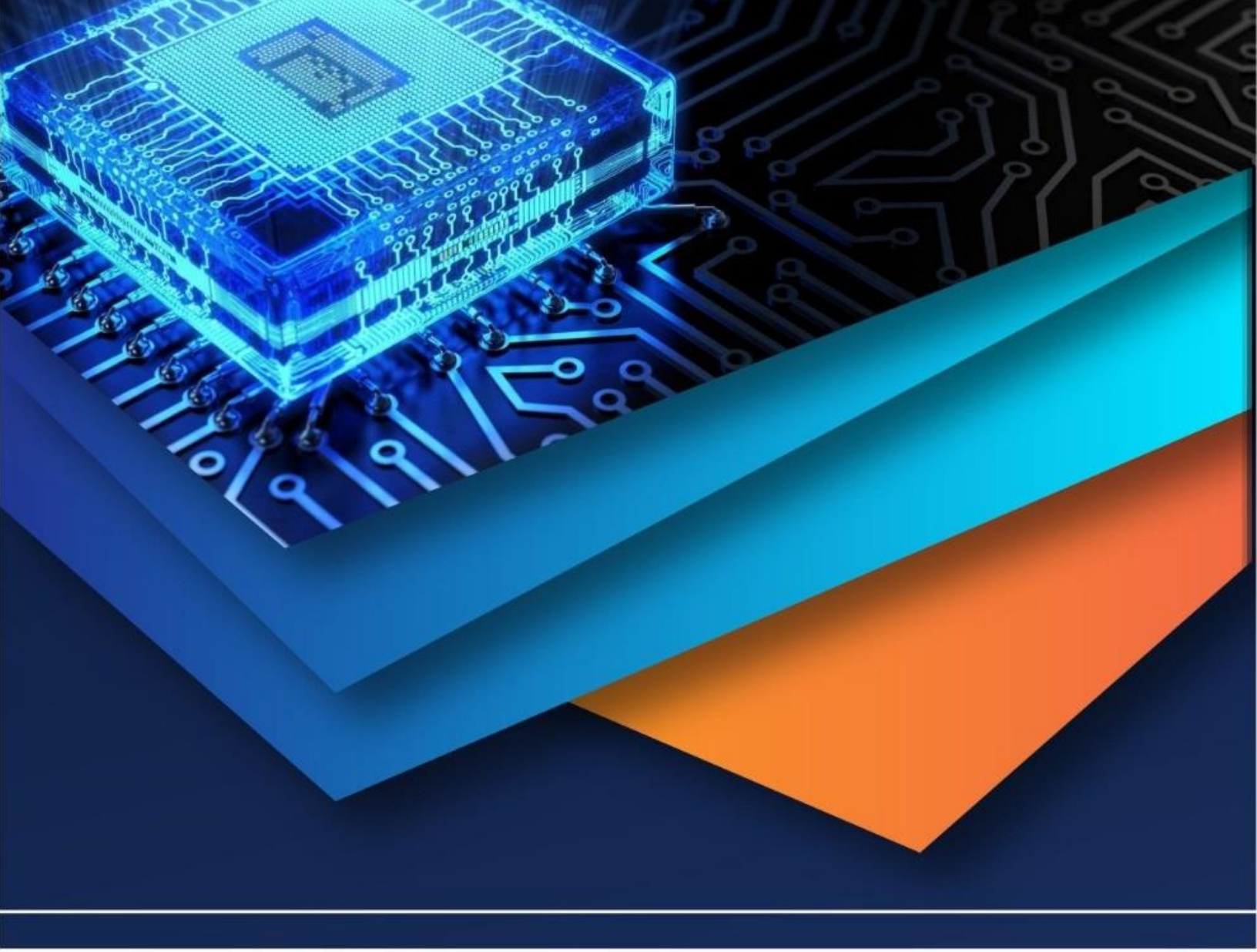
We convey our sincere thanks to all the faculties of ECM department, Sreenidhi Institute of Science and Technology, for their continuous help, co-operation, and support to complete this project.

We are very thankful to Dr. D. Mohan, Head of ECM Department, Sreenidhi Institute of Science and Technology, Ghatkesar forproviding an initiative to this project and giving valuable timely suggestions over our project and for their kind cooperation in thecompletion of the project.

We convey our sincere thanks to Dr.T.Ch. Siva Reddy, Principal, and Chakkalakal Tommy, Executive Director, Sreenidhi Institute of Science and Technology, Ghatkesar for providing resources to complete this project. Finally, we extend our sense ofgratitude to almighty, our parents, all our friends, teaching and non- teaching staff, who directly or indirectly helped us in this endeavor

REFERENCES

[1] Lev V. Utkin, Maxim S. Kovalev, and Ernest M. Kasimovc . An explanation method for Siamese neural networks . July 2017.
[2] Wael-Gomaa, Aly-Fahmy . A Survey of Text Similarity Approaches . April 2013.
[3] Xingping Dong and Jianbing Shen . Triplet Loss in Siamese Network for Object Tracking . 2017.
[4] Ming Zhong ,Pengfei Liu , Yiran Chen, Danqing Wang, XipengQiu , Xuanjing Huang . Excerptive Summarization as Text Matching . 19 Apr 2020.
[5] Jianpeng Cheng, Li Dong and Mirella Lapata .Long Short-Term Memory-Networks for Machine Reading . 20 Sep 2016.
[6] Ashish Vaswani, Noam Shazeer, Niki Parmar and Aidan N. Gomez . Attention Is All You Need . 2018.
[7] Paul Neculoiu, Maarten Versteegh and Mihai Rotaru. Learning Text Similarity with Siamese Recurrent Networks. August 2016.
[8] Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. Together We Stand: Siamese Networks for Similar Question Retrieval. August 2016.
[9] Tharindu Ranasinghe, Constantin Or˘asan and Ruslan Mitkov. Semantic Textual Similarity with Siamese Neural Networks.September 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY