iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46830

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 IX Sep 2022- Available at www.ijraset.com

Machine Translation: Spanish-to-English Translation System using RNNs

Hrithick Gokul Y[1], Dinesh N[2], Hariharan R[3]
[1, 2, 3]*Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, India*

Abstract: ***Machine translation has undergone a revolution in recent years. Older translation systems developed by linguists utilizing decades of statistical study have been superseded by new systems developed using deep learning. The core idea of a translation system will be a neural network that takes in input and generates the translation of a given sentence. This paper discusses the model which uses intermediate numerical encoding to learn the mapping from one human language to another. Two neural networks connected end-to-end provide the basis of a neural translation system. The first neural network encodes sequences of words in Spanish into an array of numbers with the same meaning. The second neural network learns to translate the numbers back into a string of words that all have the same meaning in English.***
Keywords**:** ***translation, machine translation, sequence-to-sequence, neural networks, recurrent neural networks***

I. INTRODUCTION

Language barriers can still make it difficult to get information in today's globalized world. In certain cases, it is difficult to meet the demand for translation by using solely human translators; as a result, technologies like Machine Translation (MT) are becoming more and more popular because of their capacity to solve this issue. The expanding need for solutions that enable various audiences to access content in several languages is one effect of this scenario. As a result, machine translation (MT) is highlighted as one of the tools with the greatest potential to address this issue and has received attention from both the standpoint of academic and professional contexts. Much like everything else in machine learning, machine translation is fast developing. The tools are becoming simpler to use, GPUs are becoming more potent, and there is more training data than ever before. With today's technology and software, we may construct a language translation system that is effective enough to be used in actual projects. As it turns out, deep learning has completely changed how we think about machine translation. The greatest expert-built language translation systems in the world are being defeated by simple machine learning techniques developed by deep learning researchers. This innovation's underlying technology is known as sequence-to-sequence learning. Techniques for sequence-to-sequence language translation are advancing at such a rapid rate that it is challenging to stay up. Results have greatly improved as a consequence of several recent changes (such as the addition of an attention mechanism or context tracking). We must stay up to date with new advancements as they happen if we intend to use sequence-to-sequence learning in a significant way. This study can provide useful guidance to teachers at different educational levels or areas, as well as to educators, practitioners, and researchers in the areas of personalized learning. The paper is organized as follows. Section 2 describes the method used for carrying out the systematic review of articles, Section 3 presents the related work, Section 4 describes the proposed model, Section 5 describes the whole pipeline of the project, and Section 6 concludes the paper.

II. LITERATURE SURVEY

A. Search Strategy

The following international online bibliographic databases were searched in September 2021: (a) IEEE Xplore, (b) SpringerLink, (c) ACM portal, and (d) Science direct. The search phrases used were "machine translation" AND "translation system". Searches were restricted to book chapters and the titles of articles appearing in English-language journals and conference proceedings as of the year 2020 and onwards.

B. Inclusion Criteria

The articles are chosen based on the following inclusion criteria:

1) Relevant research works proposing techniques for machine translation.
2) Discuss the methods, development, and evaluation of the translation model in detail.
3) Published in the journal from 2020-2022

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

C. Exclusion Criteria

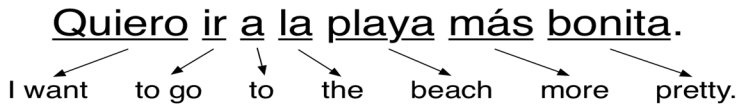
Further filtering took place by excluding the articles based on the following exclusion criteria:

1) Papers that are written in a language other than English
2) Papers to which full access was not granted
3) Papers that lacked a thorough explanation and assessment

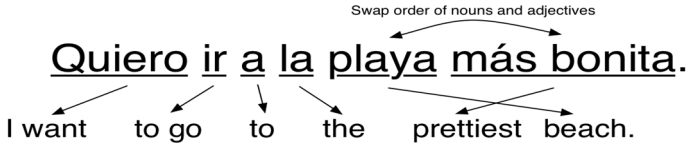
III. RELATED WORK

A. Rule-based Translation

The easiest method is to simply substitute each word in a phrase with its equivalent in the target language. This is simple to put into effect since all we need is a dictionary to check each word's translation. However, the outcomes are poor since grammar and context are disregarded.



To enhance the outcomes, language-specific rules are added as the following step. For instance, translate typical two-word sentences into a single group. Additionally, as nouns and adjectives typically appear in the opposite order in Spanish from how they do in English, the order of both is switched. Our algorithm ought to be able to translate every phrase if we simply keep adding rules until we can handle every aspect of grammar.



The early machine translation systems operated in this manner. Complex rules were developed by linguists and were coded one at a time. During the Cold War, several of the world's top linguists worked for years to develop translation systems that would make it easier to understand Russian communications. The issue is that there are no clear norms for human language. Special situations, regional variants, and outright rule-breaking abound in human languages. Who invaded whom hundreds of years ago had a greater impact on how we speak English than anybody sitting down and setting grammatical rules. Unfortunately, this only worked for straightforward papers with a basic format, like weather reports. For papers used in the real world, it was unreliable.

B. Statistical Machine Translation

If we provide statistical machine translation systems with adequate training data, they perform significantly better than rule-based systems. These concepts were enhanced by Franz Josef Och, who utilized them to create Google Translate in the early 2000s. The world had now attained access to Machine Translation. Systems that use statistical translation don't attempt to produce an accurate translation, which is their primary distinction. Instead, they create tens of thousands of potential translations and then rank them according to how likely each is to be accurate. By comparing anything to the training data, they may determine how "accurate" it is. Here's how it works:

1) First, we split our sentences into small, easily translatable parts.
2) Then, using the translations that other people have made of these similar word chunks in our training data, we will translate each of these words.
3) Finally, we will create as many phrases as we can using every combination of these parts.

Systems for statistical machine translation function effectively, but they are challenging to create and maintain. Experts must adjust and fine-tune a new multi-step translation pipeline for each new language pair we wish to translate. Trade-offs must be made since constructing these many pipelines requires so much labor. There aren't enough Georgian-to-Telugu translations occurring to warrant making a significant investment in that language combination, thus if we ask Google to translate Georgian to Telugu, it will internally convert it into English as a first step. And if we requested it to translate anything from French to English, which is the more popular option, it may use a more sophisticated pipeline.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

IV. PROPOSED MODEL

To translate text, the model employs two recurrent neural networks. The first one will be trained to encode statements in Spanish, while the second one will be trained to decode them into English. A recurrent neural network, also known as an RNN, is a slightly modified form of a neural network in which the prior state of the network is used as an input to the subsequent computation. This implies that the outcomes of current computations affect those of prior calculations. With the use of this method, neural networks may discover patterns in a stream of data. We can use it, for instance, to determine the following word in a sentence based on the first few words. RNNs are being employed more and more in many fields of natural language processing since human language is just one huge, complex pattern.

Firstly we must divide the text into sentences. If we try to feed in a full paragraph at once, our neural network will only be able to interpret one sentence at a time and will perform poorly. Though splitting text into sentences may seem simple, there are many various ways that formatting and punctuation may be nested, making it a challenging undertaking. To minimize the number of third-party libraries we need to install for this project, we'll utilize a straightforward sentence splitter built in Python. If this doesn't meet our needs, we may utilize the complex sentence-splitting models included in NLP libraries (like spaCy).

After that, text normalization will be done. The hardest aspect to do correctly, skipping here will lead to poor results in the end. Neural networks are unable to extrapolate beyond what they have observed in their training data. The neural network won't immediately understand that the word "cOMPuTERs" means the same thing as "computers" if it has never seen it before. The text has to be normalized to eliminate as much formatting variance as feasible. We'll ensure that words are capitalized consistently throughout the same sentence, repair any odd punctuation formatting, remove any strange curly quotation marks that MS Word arbitrarily placed, and more. The goal is to ensure that, regardless of how the user types a sentence, it will always be fed in the same manner.

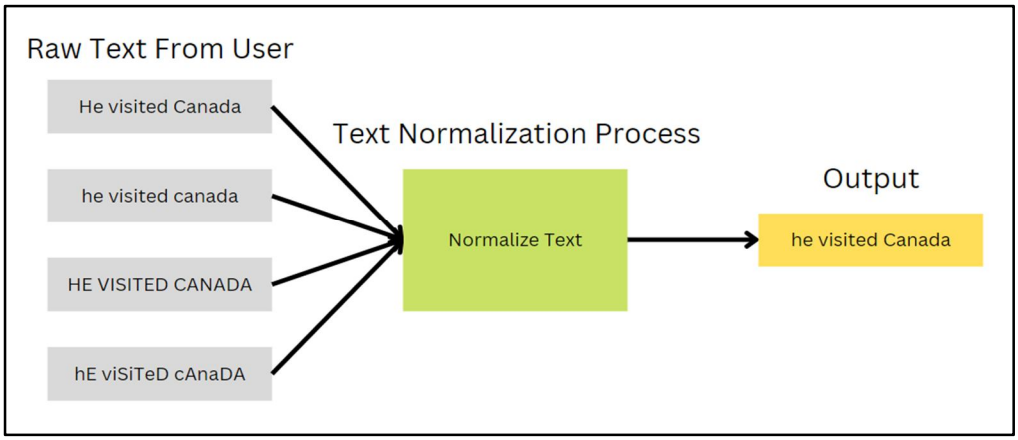


Fig. 1 Example of text normalization

After normalizing the text, we will input it into the translation model. To create the final translation, we must do the sentence splitting and text normalization procedures in reverse. So we'll have a de-normalization step and then a step to recombine the text back into sentences.

V. MODEL PIPELINE

Here is how the entire translation process will proceed, starting with the source text and concluding with the translated version:

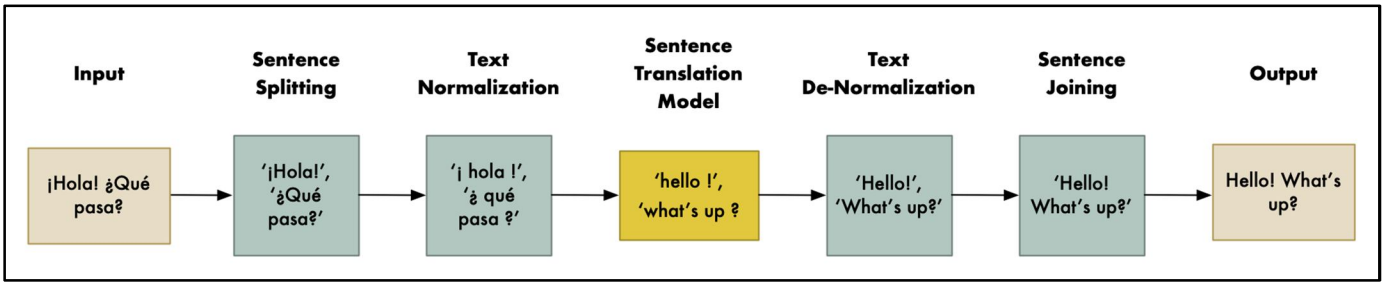


Fig.2 The Model Pipeline

A. Data Collection

To train a translation model millions of pairs of identical sentences that have been translated into two different languages are required. This is called parallel corpora. Our algorithm will learn to translate various types of text better if we have more phrase pairings. We'll require at least tens of millions of training phrases to build an industrial-strength model. Most essential, these phrases should include all forms of human communication, including slang and humor as well as professional documentation.

OPUS (Open Parallel Corpus) has already done the work of collecting sentence pairs in numerous languages. These data sources will be automatically downloaded from OPUS by the training script that we'll execute later, giving us close to 85 million translated sentence pairs. OPUS offers files according to language pairs. Therefore, by obtaining the data for that language pair instead of Spanish-to-English, we may develop, for example, a Finnish-to-Italian translation model.

language	files	tokens	sentences	bg	cs	da	de	el	en	es	et	fi	fr	hu	it	lt	lv	nl	pl	pt	ro	sk	sl	sv
bg	7,554	10.6M	0.4M		0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M
cs	9,790	15.2M	0.7M	0.4M		0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.4M	0.6M	0.6M	0.6M
da	10,316	55.8M	2.3M	0.4M	0.6M		1.9M	1.3M	2.0M	1.9M	0.6M	1.9M	2.0M	0.6M	1.8M	0.6M	0.6M	2.0M	0.6M	1.9M	0.4M	0.6M	0.6M	1.8M
de	10,254	54.9M	2.2M	0.4M	0.6M	1.9M		1.2M	1.9M	1.8M	0.6M	1.8M	1.9M	0.5M	1.8M	0.6M	0.6M	1.9M	0.6M	1.8M	0.4M	0.6M	0.5M	1.8M
el	10,242	44.1M	1.6M	0.4M	0.6M	1.3M	1.2M		1.3M	1.2M	0.6M	1.2M	1.3M	0.6M	1.2M	0.6M	0.6M	1.2M	0.6M	1.2M	0.4M	0.6M	0.6M	1.2M
en	11,199	66.8M	2.5M	0.4M	0.6M	2.0M	2.0M	1.3M		2.0M	0.6M	1.9M	2.0M	0.6M	1.9M	0.6M	0.6M	2.0M	0.6M	2.0M	0.4M	0.6M	0.6M	1.8M
es	10,397	60.9M	2.2M	0.4M	0.6M	1.9M	1.9M	1.3M	2.0M		0.6M	1.9M	1.9M	0.6M	1.8M	0.6M	0.6M	1.9M	0.6M	1.9M	0.4M	0.6M	0.6M	1.8M
et	9,776	13.2M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.7M	0.6M		0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.4M	0.6M	0.6M	0.6M
fi	10,268	40.9M	2.2M	0.4M	0.6M	1.9M	1.9M	1.2M	2.0M	1.9M	0.6M		1.9M	0.6M	1.8M	0.6M	0.6M	1.9M	0.6M	1.8M	0.4M	0.6M	0.6M	1.8M
fr	10,410	66.3M	2.2M	0.4M	0.6M	2.0M	1.9M	1.3M	2.1M	2.0M	0.6M	2.0M		0.6M	1.9M	0.6M	0.6M	2.0M	0.6M	1.9M	0.4M	0.6M	0.6M	1.8M
hu	9,718	14.7M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M		0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.4M	0.6M	0.6M	0.6M
it	10,470	59.2M	2.1M	0.4M	0.6M	1.9M	1.8M	1.2M	1.9M	1.9M	0.6M	1.8M	1.9M	0.6M		0.6M	0.6M	1.9M	0.6M	1.8M	0.4M	0.6M	0.6M	1.7M
lt	9,766	13.7M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M		0.6M	0.6M	0.6M	0.6M	0.4M	0.6M	0.6M	0.6M
lv	9,736	14.3M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M		0.6M	0.6M	0.6M	0.4M	0.6M	0.6M	0.6M
nl	10,383	59.8M	2.4M	0.4M	0.6M	2.0M	1.9M	1.3M	2.0M	2.0M	0.6M	1.9M	2.0M	0.6M	1.9M	0.6M	0.6M		0.6M	1.9M	0.4M	0.6M	0.6M	1.8M
pl	9,793	15.0M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M		0.6M	0.4M	0.6M	0.6M	0.6M
pt	10,385	61.4M	2.2M	0.4M	0.6M	1.9M	1.9M	1.3M	2.0M	1.9M	0.6M	1.9M	2.0M	0.6M	1.9M	0.6M	0.6M	2.0M	0.6M		0.4M	0.6M	0.6M	1.8M
ro	7,530	10.8M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M	0.4M		0.4M	0.4M	0.4M
sk	9,740	15.1M	0.7M	0.4M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.4M		0.6M	0.6M
sl	9,703	14.6M	0.6M	0.4M	0.6M	0.6M	0.5M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.6M	0.4M	0.6M		0.6M
sv	10,345	51.8M	2.3M	0.4M	0.6M	1.9M	1.8M	1.3M	1.9M	1.8M	0.7M	1.9M	1.9M	0.6M	1.8M	0.6M	0.6M	1.9M	0.7M	1.8M	0.4M	0.6M	0.6M	

Fig.3 A list of language combinations of the European Parliament provided by OPUS

B. Data Pre-processing

Two text files are downloaded with every data source from OPUS. One text file has a list of sentences in English, while the other contains a list of sentences in Spanish, both in the same sequence. The result will be a large number of text files, so we need to go through several steps to prepare the training data.

1) *Combine:* Create a single, enormous text file from all the English text files. Similar to the English files, merge all the Spanish text files, being careful to maintain the sequence of all the phrases so that the English and Spanish files match.
2) *Shuffle:* While retaining the sentences in the English and Spanish files in the same relative order, randomize the order of the sentences in each file. Instead of learning to translate formal data first, then informal data second, and so on, mixing training data from various sources will enable the model to learn to generalize across various sorts of text.
3) *Split:* Split the master data files into training, development, and testing portions for each language. The majority of the training data will be used to train the model, but some phrases will be withheld so that the model may be tested and ensured to be functional. This guarantees that the model is tested using words that it was never exposed to during training, allowing us to be certain that it didn't just remember the right responses.
4) *Subword Segmentation:* Given that new words are coined regularly, it is impossible to get training data that includes every conceivable phrase. We are going to apply a technique called subword segmentation to assist our translation system in handling terms that have never been encountered before. The specific implementation of subword segmentation will we use is called BPE, or Byte Pair Encoding.

C. Translation Model

The model's general structure is as follows, The encoder at each time step receives a single element from the input sequence, processes it, gathers data for that element, and propagates it forward. The intermediate vector is the model's final internal state as created by the encoder. It includes details about the whole input sequence to aid the decoder in making precise predictions. Given the entire sentence, the decoder predicts an output at each time step.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

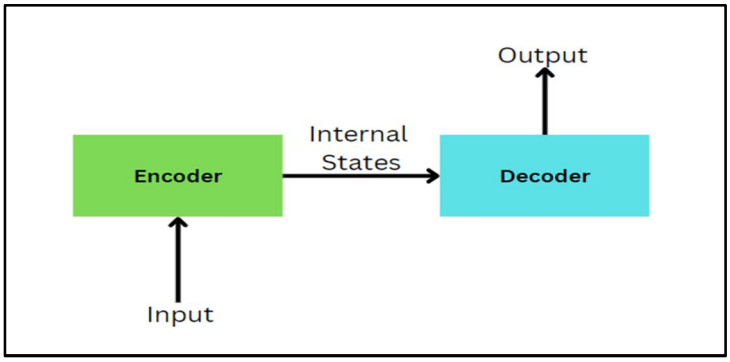


Fig.4 The Structure of the Translation Model

1) Encoder Part

LSTM/GRU cells essentially make up the encoder. When an encoder receives an input sequence, it stores the data as internal state vectors. Only internal states are used, and the encoder's outputs are discarded. If the input sequence is m characters long, it will take m time steps for LSTM to read the full sequence because it only reads one element at a time. The input at time step m is Xm. At time step m of the LSTM, the internal states hm and cm are present. There is just one internal state hm for the GRU. The result at time step m is Ym.

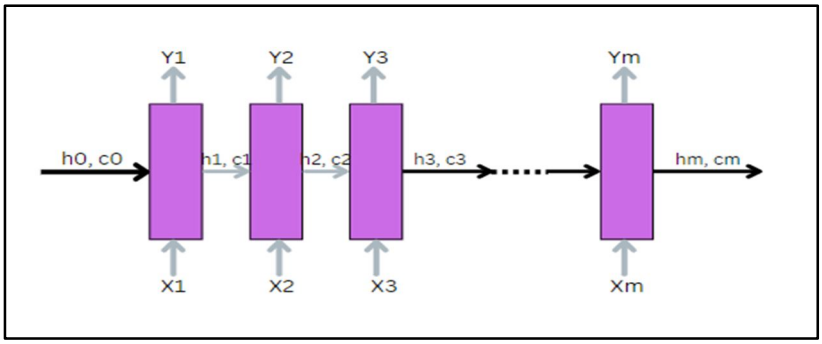


Fig. 5 LSTM for encoder

Using word embedding, which turns each word into a vector of fixed length, each Xm (each word) is represented here as a vector. Getting to the interior states now (hm, cm). The LSTM's readings up to time step m are learned. Ho and Co are initialized either randomly or with zeroes (both are vectors). The number of units in the LSTM cell is the same as the dimension of ho and co. The core of the whole input sequence is included in the final state, H4, C4. The predictions made by the LSTM at each time step are the output Ym. In situations involving machine translation, we provide the results after reading the complete input sequence. As a result, we reject Ym at each time step in the encoder since it is useless.

2) Decoder Part During Training

Unlike the encoder component of the model, which operates in the same way during training and testing, the decoder has a separate operating process. We will add START_ at the beginning of the output sequence and _END after the output sequence for the decoder to distinguish the starting and end of the sequence.

The decoder's beginning states (ho, co) are set to the encoder's end values. It may be said that the encoder gathers the data, and the decoder is taught to produce the output based on that data. For the decoder to begin producing the following word, we must first enter the START_. Additionally, we train the decoder to anticipate the _END following the final word in the Spanish text. Here, we employ the teacher-forcing method, in which the actual output, rather than the projected output from the previous time step, is input at each time step. To update the model's parameters, the mistakes are backpropagated over time and the loss is computed on the projected outputs from each time step. The decoder's last states are ignored since we already have the output, therefore they serve no purpose.

Here is a visual representation of the working of the decoder:

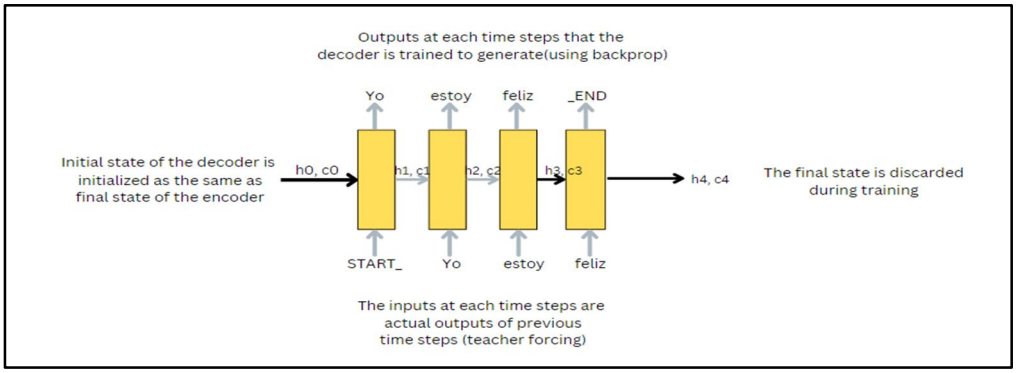


Fig. 6 Decoder LSTM at training

3) Decoder Part During Testing

The end states of the encoder are used as the decoder's beginning states. Each time step in the decoder's LSTM processes a single word. The START_ is the first character that enters the decoder. The internal states produced after each time step are sent into the subsequent time step as the beginning states. for e.g, The internal states generated after entering START_ are provided as the beginning states at t=1 and t=2, respectively. Each time step's output is used as input for the following time step. When the decoder anticipates the END_, we are informed that the sequence has ended.

VI. RESULTS

The opening paragraph of the first Harry Potter book's Spanish translation serves as the sample input. The output looked like this:

```
Input:
El niño que vivió.
El señor y la señora Dursley, que vivían en el número 4 de Privet
Drive, estaban orgullosos de decir que eran muy normales,
afortunadamente.

Output:
The boy who lived.
Mr. and Mrs. Dursley, who lived at number 4 on Privet Drive, were
proud to say they were very normal, fortunately.
They were the last people who would expect to find themselves related
to something strange or mysterious, because they weren't for such
nonsense.
```

Fig. 7 Result

Despite some differences, our translation conveys the same concept as the book's English version. And our translation more closely matches the Spanish text.

VII. CONCLUSION AND FUTURE SCOPE

The amount of training data we have and the computing power we have access to are the main constraints on this method. This already outperforms statistical machine translation systems that took two decades to create. It is not necessary to be familiar with any linguistic rules for this. These rules are created by the algorithm itself. As a result, we don't need professionals to fine-tune our translation workflow at every stage. That is done for us by the computer. Almost any sequence-to-sequence problem may be solved with this method. It turns out that many intriguing issues are problems of sequence to sequence. Techniques for sequence-to-sequence language translation are advancing at such a rapid rate that it is challenging to stay up. There aren't even Wikipedia pages for many of the newest advances (such as adding an attention mechanism or tracking context) even though they are greatly boosting results. We must stay up to date with new developments as they happen if we intend to use sequence-to-sequence learning seriously. This study assists the user in developing and testing a fundamental translation system with crucial features. This aids in understanding the principles of the encoder-decoder model used in translation and piques the user's interest in further research in the area.

REFERENCES

[1] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A Survey of Multilingual Neural Machine Translation. ACM Comput. Surv. 53, 5, Article 99 (September 2021), 38 pages. https://doi.org/10.1145/3406095

[2] J. Liu, "Multimodal Machine Translation," in IEEE Access, doi: 10.1109/ACCESS.2021.3115135.

[3] B. Zhang, D. Xiong, J. Xie and J. Su, "Neural Machine Translation With GRU-Gated Attention Model," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4688-4698, Nov. 2020, doi: 10.1109/TNNLS.2019.2957276.

[4] G. Tiwari, A. Sharma, A. Sahotra and R. Kapoor, "English-Hindi Neural Machine Translation-LSTM Seq2Seq and ConvS2S," 2020 International Conference on Communication and Signal Processing (ICCSP), 2020, pp. 871-875, doi: 10.1109/ICCSP48568.2020.9182117.

[5] Rivera-Trigueros, I. Machine translation systems and quality assessment: a systematic review. Lang Resources & Evaluation 56, 593–619 (2022). https://doi.org/10.1007/s10579-021-09537-5

[6] Alanoca, H.A., Chahuares, B.A., Caceres, K.A., Saire, J.C. (2023). Neural Machine Translation for Aymara to Spanish. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems, vol 544. Springer, Cham. https://doi.org/10.1007/978-3-031-16075-2_19

[7] Haque, R., Liu, CH. & Way, A. Recent advances of low-resource neural machine translation. Machine Translation 35, 451–474 (2021). https://doi.org/10.1007/s10590-021-09281-1

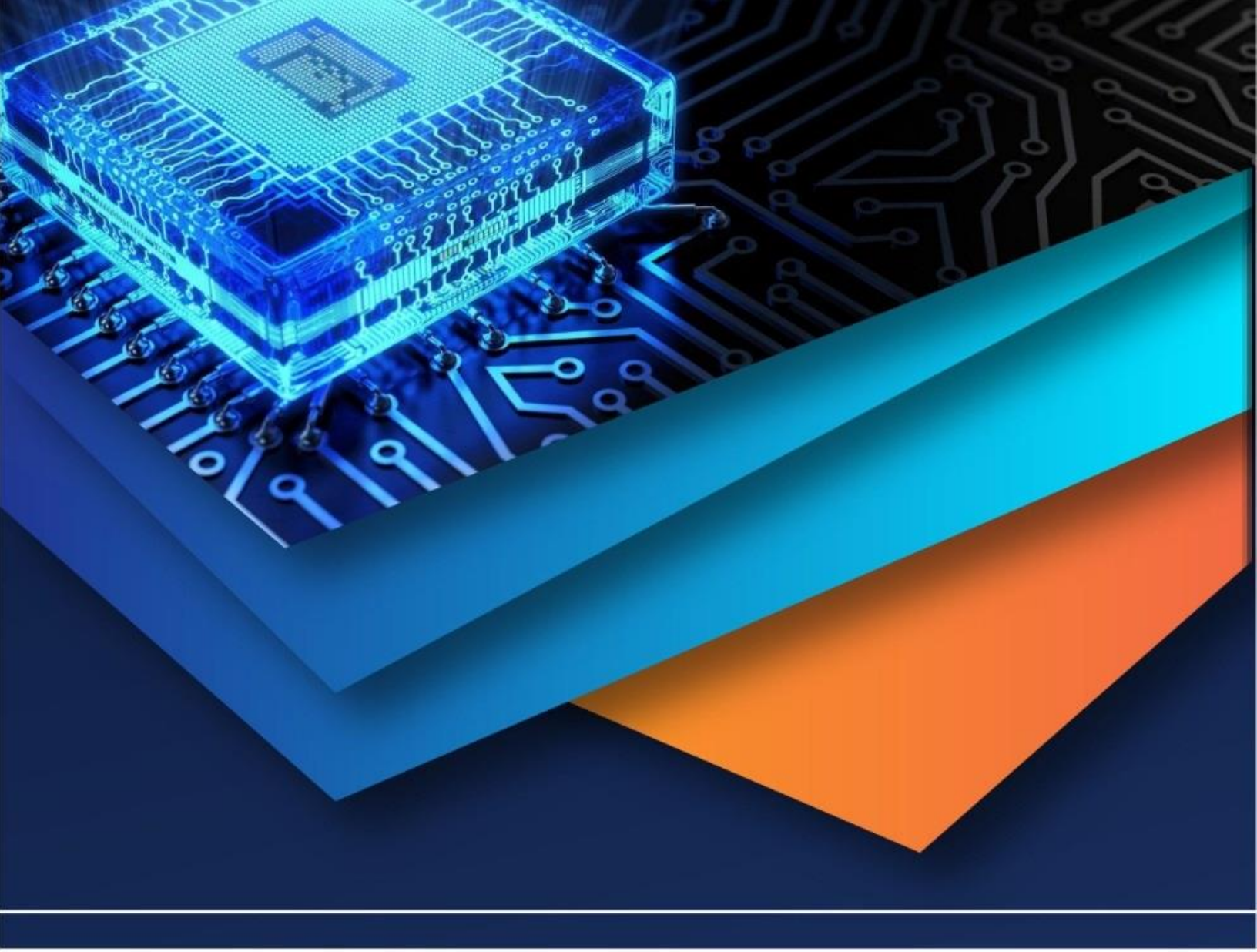
[8] Zhang, J., Zong, C. Neural machine translation: Challenges, progress and future. Sci. China Technol. Sci. 63, 2028–2050 (2020). https://doi.org/10.1007/s11431-020-1632-x

[9] Ruíz, S., Providel, E., Mendoza, M. (2021). Fake News Detection via English-to-Spanish Translation: Is It Really Useful?. In: Meiselwitz, G. (eds) Social Computing and Social Media: Experience Design and Social Network Analysis . HCII 2021. Lecture Notes in Computer Science(), vol 12774. Springer, Cham. https://doi.org/10.1007/978-3-030-77626-8_9

[10] Gandhi, D., Mehta, J., Bari, P. (2022). Ablation Analysis of Seq2Seq Models and Vanilla Transformers for Spanish to English Translation. In: Rout, R.R., Ghosh, S.K., Jana, P.K., Tripathy, A.K., Sahoo, J.P., Li, KC. (eds) Advances in Distributed Computing and Machine Learning. Lecture Notes in Networks and Systems, vol 427. Springer, Singapore. https://doi.org/10.1007/978-981-19-1018-0_60

[11] Jena, O.P., Tripathy, A.R., Patra, S.S., Chowdhury, M.R., Sahoo, R.K. (2022). Automatic Text Simplification Using LSTM Encoder Decoder Model. In: Sahoo, J.P., Tripathy, A.K., Mohanty, M., Li, KC., Nayak, A.K. (eds) Advances in Distributed Computing and Machine Learning. Lecture Notes in Networks and Systems, vol 302. Springer, Singapore. https://doi.org/10.1007/978-981-16-4807-6_23

[12] Makin, J.G., Moses, D.A. & Chang, E.F. Machine translation of cortical activity to text with an encoder–decoder framework. Nat Neurosci 23, 575–582 (2020). https://doi.org/10.1038/s41593-020-0608-8



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY