Title Generation Using Lstm With Beam Search For Enhancing The Contextually Relevant Headlines

[1]Dr.G. Dona Rashmi , [2]Dr.O.P. Uma Maheswari

[1]Assistant Professor, Department of Computer Applications,
Kongunadu Arts and Science College, Coimbatore, donamca08@gmail.com
[2]Associate Professor, Department of Computer Science,
P.K.R. Arts College for Women, Gobichettipalyam, umapritika@gmail.com

Abstract

Title Generation employing NLP is a cutting-edge technology in natural language processing that automatically creates compelling and contextually suitable headlines or titles for articles, blog entries, news items, and social media posts. This research looks at NLP title generation models and methodologies. Title Generation Using NLP Models is a comprehensive study that investigates strategies for automatically developing engaging and contextually relevant titles for various content genres. Spacy, RNN, LSTM, and LSTM with Beam Search NLP models are investigated in this work. The first model uses the NLP application to preprocess tokenization, named entity recognition, and part-of-speech tagging. The research demonstrates t enhance title generation pipelines. The second model investigates Vanilla RNN, a basic recurrent neural network. Despite their simplicity, vanilla RNNs is an excellent introduction to sequential data processing. The study investigates the benefits and cons of Vanilla RNN's title generation capabilities. The third model employs LSTM, an improved RNN architecture that addresses the vanishing gradient problem while still capturing sequence dependencies. Because of its long-term memory, LSTM is suitable for title development. The research compares LSTM with Vanilla RNN. In the fourth model, LSTM and Beam Search increase title diversity and relevancy. Beam Search efficiently explores

numerous candidate sequences and selects the titles with the highest likelihood based on beam width. The research investigates how Beam Search influences the quality of LSTM title creation. Throughout the study, each model is assessed using a variety of datasets and criteria. The article also investigates and improves real-world NLP-based title generation applications. This research sheds light on how to use NLP models for title development in order to improve the engagement and success of digital media platforms, information retrieval tools, and content marketing strategies.

Keywords: Title Generation, natural language process, LSTM, Beam search, Vanilla RNN.

I. INTRODUCTION

The skill of producing intriguing and contextually appropriate headlines has never been more important in the digital age, when information is copious and attention spans are short [1]. A well-crafted title may be the difference between grabbing an audience's attention or falling into oblivion whether it comes to news stories, blog entries, social media updates, or any other kind of information [2]. This study looks into the world of title production using cutting-edge Natural Language Processing (NLP) methods, with the goal of creating headlines that not only stimulate attention but also properly reflect the information they represent [3]. The union of two strong components is at the center of this endeavor: LSTM (Long Short-Term Memory), an advanced recurrent neural network architecture noted for its sequence modeling abilities, and Beam Search, a dynamic search method [4]. They produce a powerful synergy, with the potential to increase both the variety and relevancy of generated titles [5]. This research investigates the integration of LSTM with Beam Search, revealing its ability to improve the quality of contextually relevant headlines across several content genres [6]. We embark on a journey to discover the transformative possibilities of this novel approach in title generation through empirical evaluation and practical applications, ultimately aiming to

provide a valuable resource for content creators, marketers, and information disseminators in the digital landscape [7].

In today's digital age, when information floods our screens and online diversions abound, the art of crafting engaging and contextually precise headlines has taken on new importance. These headlines serve as the entry point for our interaction with material, whether it's news stories, blog entries, or social media snippets [8]. A well-crafted title may act as a beacon, attracting readers or viewers and ensuring that a piece of content stands out among the digital noise [9]. In response to this changing environment, this study attempts to negotiate the complicated art of title creation by using the tremendous capabilities of Natural Language Processing (NLP) approaches. At its foundation, title creation is a search for brevity and impact, attempting to condense the meaning of extensive information into a few carefully selected words [10]. However, given the complexities of language and the subtleties of context, this apparently straightforward endeavor is anything but. To address this issue, we dig deep into the domain of natural language processing (NLP), where cutting-edge technology and machine learning algorithms have started to excel at developing titles that not only pique the reader's interest but also successfully contain the material they represent [11]. The intersection of two remarkable components lies at the heart of our investigation: LSTM, an advanced recurrent neural network architecture renowned for its prowess in modeling sequences and capturing temporal dependencies, and Beam Search, a dynamic search algorithm that adds diversity and relevance to generated titles [12]. Together, these two factors generate a potent combination that has the potential to transform the way we create and communicate headlines [13]. Our research aims to debunk the myths surrounding the integration of LSTM with Beam Search and reveal its exceptional potential in improving the quality of contextually relevant headlines across all content types. We go on an empirical journey, systematically assessing the performance and effect of this unique technique across a range of datasets and real-world applications [14]. The objective is simple: to discover transformational possibilities those goes beyond academic curiosity and immediately assist content producers,

marketers, and information disseminators in the ever-changing digital ecosystem [15]. In a world where information competes for milliseconds of our ephemeral attention, the ability to develop interesting and accurate titles is more than simply a scholarly endeavor; it's a critical driver of content exposure, user engagement, and successful communication [16]. With Beam Search, we aim to provide a valuable resource, offering insights and tools that empower individuals and organizations to navigate the digital realm more effectively, ensuring their messages rise above the digital noise and resonate with their intended audience [17].

II. BACKGROUND STUDY

I.C. Irsan et al. [5] With AUTOPRTITLE, we show how to generate titles for pull requests automatically. Our web software requires developers to manually input the URL for the pull request once it has been made. They may also link to the relevant problems and offer a brief description of the pull request in the web tool. Our tool will produce a proposed PR title when you click the Generate PR Title button. After that, programmers may simply paste the pull request's title onto the new page. AUTOPRTITLE employs the cutting-edge summarization technique known as BART. Our findings show that BART is superior to the status quo. Additionally, reviewers believe that AUTOPRTITLE is simple to use and helpful for quickly producing high-quality titles for pull requests. Future work on AUTOPRTITLE will include incorporating more sources of data, such as developer comments on proposed modifications to the code.

K. Kaku et al. [6] these authors research provide the prototype of an extractive document title generating system based on BERT to aid authors who are not yet conversant with the conventions of academic paper titling. Based on our experiments, we found that the title assessment model performed better than Japanese graduate students in judging the quality of paper titles. The grammar checker also helped reduce potential paper titles that weren't essential without compromising on quality.

M. Mohseni and H. Faili [9] the author focuses on two tasks specific to Persian, a low-resource language: title creation and key phrase extraction. To train our full-stack model, we

collect a large corpus of Persian scholarly works. Our analysis of LSTM encoder-decoder networks' sensitivity to input length shows how well it can mimic long Persian texts. The great degree of derivation and inflection in Persian allows for the creation of words that have a common etymological basis (the "root" or "lemma"). We investigate BPE, an approach to processing at the subword level, and evaluate the outcomes alongside those of processing at the word level. We propose a method for key phrase extraction in which the model is taught to produce a single key phrase at a time during training but is then taught to extract a set of n best key phrases during inference. This technique has the potential to increase ROUGE-1 by 10%, ROUGE-2 by 5%, and F1 by 2%.

M. Riku and K. Masaomi [10] In this research, we provide an image captioning technique for automatically creating product titles given a picture of the product. Furthermore, we combined two prior works—the Semantic Compositional Network and top-down attention—to propose a novel approach to picture captioning. The results of these experiments reveal that the image captioning approach is quite effective at generating titles for products based on their images. Additionally, they verify that the integrated design improves assessment metrics.

T. -F. Hsieh et al. [14] In this research, we provide a GAN-based title generator that does not need paired training data. Our n-gram discriminators help produce content that is both useful and legible by humans. Positive outcomes have been obtained experimentally using our model. Our long-term goal is to enhance the performance of the summarization pipeline by using contextualized embeddings like BERT. Potential future directions include expanding the suggested model's use beyond the news domain to include additional sources.

X. Ma et al. [16] This research combined an encoder-decoder model with a copy mechanism to create ATTSUM, a deep attention-based summarizing model that may help with this issue. The architecture of the model was effective in capturing the underlying semantic information and long-range dependency present in report bodies despite the physical distance between words. The copy process was introduced so that unusual words might be cherry-picked

for use in the final titles. There were 333,563 unique bug reports utilized in the tests. Findings from automated evaluation confirmed the result obtained by human review, showing that ATTSUM performed better than state-of-the-art approaches.

III. MATERIALS AND METHODS

The "Materials and Methods" section of a research paper is the cornerstone of scientific inquiry, providing a detailed roadmap for how experiments, investigations, and data collection were conducted. This section serves as a critical guide for other researchers seeking to replicate or build upon the study's findings, ensuring transparency and rigor in the scientific process.

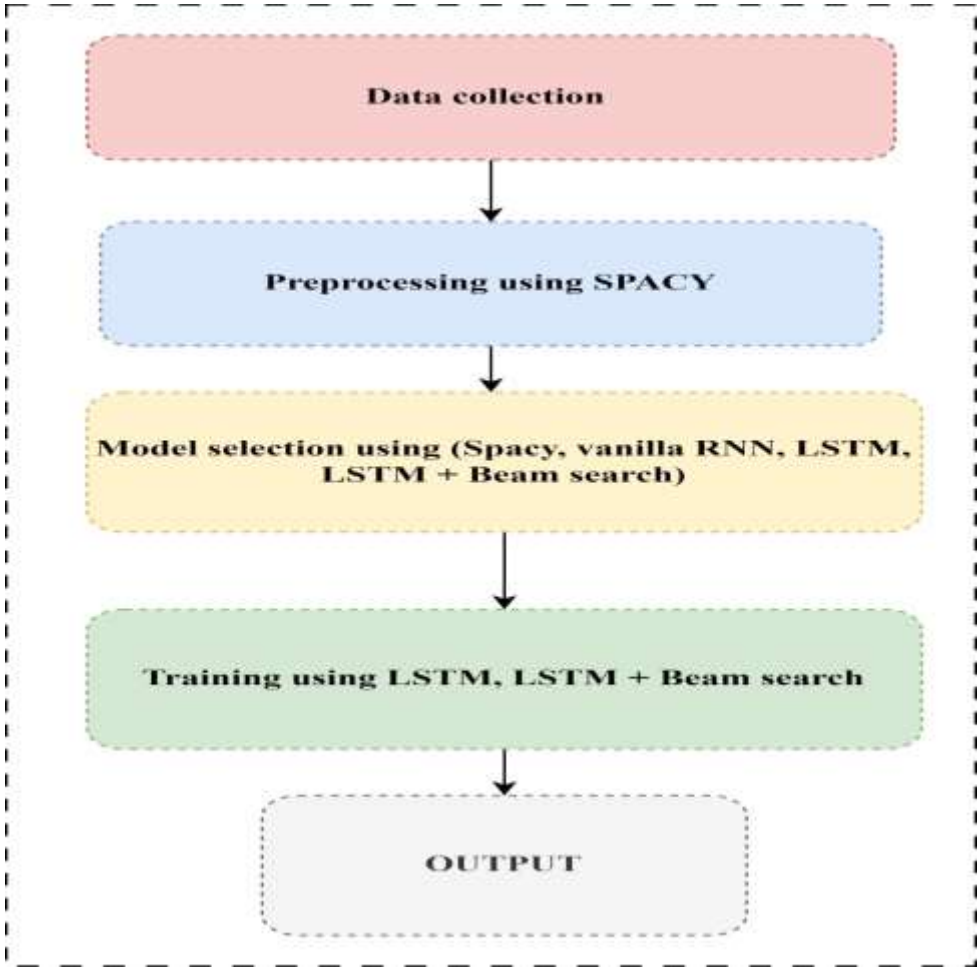

Figure 1: Block diagram

3.1 Preprocessing using SPACY

SpaCy has extensive language and text processing capabilities, and it is a highly effective and versatile natural

language processing (NLP) library E. Partalidou et al. (2019). With features including tokenization, part-of-speech tagging, named entity identification, and dependency parsing, SpaCy is a versatile NLP tool that is both user-friendly and powerful. Users may rapidly and reliably extract relevant insights from textual data using its pre-trained models, which support many languages. SpaCy is a popular option among academics, developers, and companies looking for powerful and efficient NLP solutions for tasks like text analysis, information extraction, and language comprehension because to its modular design, speed, and accuracy.

Although first created for NLP problems, its use has expanded greatly. Data categorization using the Naive Bayes Classifier is an entirely probabilistic process. Under the assumption that each pair of attributes is conditionally independent, Bayes' theorem is used to predict the text label. Using the Bayes theorem (Equation 8), one may determine the probability that a given document will be assigned to a certain outcome category.

$$\mathrm{P}\left(\frac{\mathrm{co}}{\mathrm{cd}}\right) = \frac{\mathrm{P}\left(\frac{\mathrm{cd}}{\mathrm{co}}\right) * \mathrm{P}(\mathrm{co})}{\mathrm{P}(\mathrm{cd})} \text{ ------ (1)}$$

Among its many applications is text analysis, where the naive Bayes classifier shines. The technique of assigning labels to texts based on their contents is known as text classification. The word "classification" encompasses both the act of classifying texts and the act of labeling them.

The term "relational data extraction" (RE) refers to the process of extracting data concerning relationships between items in a text. The importance of relation extraction increases when dealing with data from unstructured sources, which commonly takes the form of unstructured language. Having access to a company's internal semantic links is useful in many different areas, including information extraction, language learning, and knowledge discovery. Equation 9 provides the mathematical expression for this association.

$$\mathrm{t} = (\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{n}) \text{ ------ (2)}$$

where ei represents the parties to a connection R that has already been established in a document D.

For example, in the phrase "NLP implemented in logistics," the conjunction "implemented in" establishes

causality between the two terms. In order to record and monitor this association, we use triples (NLP, utilized by, Logistics). The first method may be used to discover connections between objects, and it is transparent and may extract important information directly from raw text with little involvement. Using supervised information extraction, which draws on prior knowledge, may achieve the same goal.

3.2 Model selection using Vanilla RNN (Recurrent Neural Network)

Vanilla RNN (Recurrent Neural Network), the simplest kind of neural network design, excels at handling sequential input A. Joshi et al. (2022). RNNs have an advantage over feedforward neural networks, which analyze input in a single pass without keeping any of the context from preceding iterations because of their loops, which allow them to store information from prior time steps in a sequence in a hidden state. Natural language processing, speech recognition, and time series prediction are just few of the areas where RNNs shine thanks to their capacity to model temporal connections and contextual information utilizing this memory. However, Vanilla RNNs have drawbacks, the most significant of which is the vanishing gradient issue, which limits how well they can capture long-range relationships. Training issues and perhaps subpar results on tasks needing large amounts of context are common results of this problem. Recent RNN architectures like as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are often employed for sequence modeling problems in state-of-the-art deep learning applications due to these limitations.

Input, recurrent hidden, and output is the three components that make up a basic RNN. There are N inputs to this layer. This layer takes as inputs time-varying vectors. The connections between the input units and the hidden units in the hidden layer of a fully connected RNN are described by the weight matrix WIH. Figure 1b demonstrates the recurrent connections between M hidden units in the hidden layer. There is some speculation that giving hidden units non-zero values to start with might

improve network throughput and stability. The state space or "memory" of the system is what the hidden layer calls it.

$h_t = f_h(o_t)$ ----- (3)

Where

$o_t = W_{IHX_t} + W_{HH}h_{t-1} + b_h$ ------- (4)

The hidden layer's activation function, fH(•), and the hidden units' bias vector, bh, are denoted respectively. The hidden units are linked to the output layer by weighted connections WHO. yt = (y1, y2,..., yP) is the formula for determining the output units of the P-th layer.

$y_t = f_o(W_{HO}h_t + b_o)$ ------ (5)

In the output layer, the activation function fO(•) must be distinguished from the bias vector Bo. Since the input-target pairings occur in a certain sequence, the aforementioned steps are repeated indefinitely over t = (1,..., T). An RNN is shown to be made up of recursive nonlinear state equations by (1) and (3). The hidden states make a prediction about the output vector from the input vector at each timestep. A recurrent neural network's (RNN) hidden state is a set of values that represents, in a unique fashion, the network's state history over several time periods. With the information gathered, the network's future behavior may be specified and reliable predictions can be made at the network's output layer. Each unit's activation function in an RNN is straightforwardly nonlinear. With enough training, a basic model could be able to pass for a more complex one.

3.3 Training using long short-term memory and LSTM with Beam Search

3.3.1 Long short-term memory

The vanishing gradient issue in conventional RNNs is solved by the Long Short-Term Memory (LSTM) architecture, which is able to successfully capture long-range relationships in sequential data T. Cai et al. (2022). Cells in LSTMs have a state, a hidden state, and three gates (forget, input, and output) that control the flow of data. Natural language processing, speech recognition, time series prediction, and autonomous driving are just some of the areas where LSTMs may be put to good use because of their ability to remember

and selectively update information over time. Since LSTMs allow for more robust modeling and prediction of complicated sequences, they have become a staple in many machine learning applications that deal with sequential data.

Spectrum sensing uses cyclostationary detection, one of the most trustworthy feature extraction techniques. A signal's underlying cyclic features may be inferred by estimating its cyclic profile. Depending on their physical features, including carrier frequency and symbol speeds, communications signals display a variety of cyclic qualities. Communications signals may be sorted into several classes based on their cyclic characteristics, as seen in.

The spectral correlation function $S_x^a(f)$ for a cyclostationary process x(t) is the Fourier transform of the cyclic autocorrelation function R_x^a for x(t):

$S_x^a(f) = \int_{-\epsilon}^{\epsilon} R_x^a(T) e^{-j2\pi f t_{dT}}$ ------- (6)

where c and t represent the period between cycles. The auto coherence function magnitude, denoted by, provides a normalized form of the SCF by:

$|C_x^a(f)| = \frac{|S_x^a(f)|}{\sqrt{S_x^0(f+a/2) S_x^0\left(f-\frac{a}{2}\right)}}$ ------- (7)

By optimizing the auto coherence function throughout the whole frequency spectrum, we can define the cyclic profile $I_x(a)$ for each cyclic frequency:

$I_x(a) = \max_f |C_x^a(f)|$ -------- (8)

For this comparison, we will not be feeding the LSTM layer the discrete samples x[k], but rather the cyclic profile Ix(). This model, the cyclostationary-based LSTM model, will be distinguished from the raw data-based model by how it handles the discrete samples x[k].

3.3.2 Beam Search

Beam Search is a popular search technique used in NLP and ML for sequence-generating applications like machine translation and text synthesis C. Liu et al. (2022). At each decoding stage, it keeps a "beam" of the highest-scoring candidate sequences, scores them using a specified scoring function, and then prunes less promising possibilities to control computing complexity. In huge search areas, this algorithm is a good option since it achieves a compromise

between variety and quality. Beam Search is an important sequence decoding method that, although it may not always provide the best results, serves a critical purpose in many contexts where it is necessary to generate coherent and contextually appropriate sequences.

3.3.3 LSTM with Beam Search

Long Short-Term Memory (LSTM) is a specific recurrent neural network architecture that has been combined with the Beam Search algorithm to improve sequence generating tasks, especially in natural language processing G. Szűcs and D. Huszti (2019). Beam Search is a search strategy for effectively examining several candidate sequences, whereas LSTM is great at capturing long-term relationships in data. The sequence continuations in this hybrid method are generated using a Long Short-Term Memory (LSTM) network, and then the most promising possibilities are culled and chosen using Beam Search's probabilistic scoring system. Applications such as machine translation, text creation, and voice synthesis benefit from this cooperation because of the increased variety and contextual significance of the created sequences.

Previous beam-search decoding methods have been criticized for focusing only on historical data. At first glance, relying on data from the future seems to be a violation of causality. How can we use something that hasn't been expressed to interpret what we've read? Human readers, however, do just that. The meaning of words and often even whole phrases is revealed only when seen in the context of what comes after. In certain cases of sequence prediction, the usage of bidirectional LSTMs may provide improved results.

We need the Bidirectional Beam Search (BBS) to consider temporal dependencies in both directions for this project to succeed. We take this into account on the assumption that fusing historical and prospective data improves the model's performance when creating the summary. This method is similar to how humans summarize information. In which he or she reads the whole text backwards and forwards in an effort to find the most relevant details while staying within the bounds of a fixed summary length (compression rate).

Knowing the whole sequence is the biggest obstacle to successfully implementing a bidirectional decoder. Something tricky during the trial run to get around this problem, we do a one-way backward beam search to get the whole reversed sequence at the outset:

$$B_{[1:T_y]} = \text{topk} \sum_{i=1}^{T_y} \text{logp}\left(y_i \middle| Y_{[i+1:T_y]}\right) \text{-------- (9)}$$

$$Y_{[1:T_y]} = \sum_{i=1}^{T_y} (r \sum_{i=1}^{t} \log p\,(y_i | Y_{[1:i-1]}) \text{-------- (10)}$$

Algorithm 1: LSTM with Beam Search

Input:

Bidirectional Beam Search Parameters: These parameters include the beam width, scaling value (γ), and any other relevant hyperparameter that control the algorithm's behavior.

Steps:

 Initialization: Initialize the beam with candidate sequences.

 LSTM **Processing**: Use bidirectional LSTM to process the input sequence (X) and capture temporal dependencies in both directions. This step may involve forward and backward passes through the LSTM.

 Beam **Search**: Use Beam Search to generate sequence continuations. At each step, calculate the probability scores for possible candidates based on the model's output and the context.

 Candidate **Selection**: Select the most promising candidates based on their probability scores, considering both forward and backward dependencies.

Output:

- **Generated Sequence (Y)**: This is the sequence that your algorithm generates as the output. It could be a sequence of words, tokens, or any other relevant units depending on the specific task.

IV. RESULTS AND DISCUSSION

Dataset Used	
Dataset	**Description**
BBC news data (Business)	2225 documents and 737 documents from the BBC news and Sports website respectively.
Medium tech articles	This dataset was scraped from Medium and contains the summary and title of the article.
Department of justice press	13,087 press releases which are scraped from data.gov site of the USA using beautiful soup.
All news dataset	Archive.org is used to scrape data from fox news, CNN etc. using beautiful soup

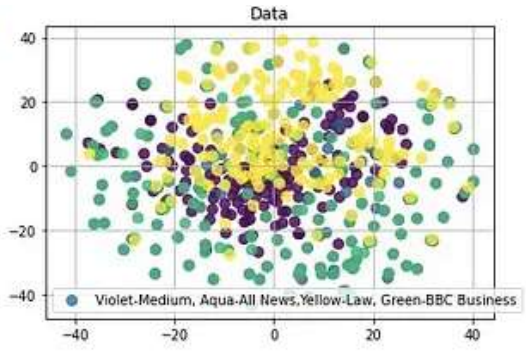

Figure 2: Data set Description, Data Distribution

The figure is created by using the 200 most commonly appearing terms in each dataset's title. It demonstrates that frequent terms are close together, whereas rare words are spread apart, indicating that our datasets are relatively generic. As a consequence, our model may be made broader by including directions predictions from other sources. Results for individual datasets and the aggregated one, consisting of around 1 lakh articles and headlines, were both presented.

The following models were used:

Spacy model: In this model, we employed a mix of frequency-based prioritization, POS tagging (noun + adjective), and title-based phrase selection. The model's lack of sophistication and inability to teach us anything meant that we quickly moved on to a more advanced model.

4.1 Vanilla RNN:

We utilize the tensor flow library to implement basic RNN, which is effectively a fully-connected RNN in which the output from one timestep is passed on to the next. Because it just takes into account the most recent input and ignores all others, it works well with our data. It is the basis on which new models may be built.

4.2 LSTM:

We use LSTM equipped with an encoder-decoder architecture to achieve our goals. The encoder makes use of a reversible long short-term memory (LSTM) layer, reading one word at a time and retraining the hidden layer depending on the previous word and the current word. One

word at a time, the decoder creates a summary using a unidirectional LSTM layer.

4.3 LSTM with Beam Search:

To go on with LSTM, we employ beam search, which picks the best alternatives based on beamwidth; among the best available alternatives, only a few number are selected that are both optimal and significantly effective.

We employed three assessment measures to test our output, which are mentioned below:

- The BLEU score: Another name for this is Bilingual Evaluation. The score for novices takes into account the number of n-grams of varying lengths that have the model's projected heading.
- Google word2vec distance similarity: To measure how diverse two texts are, this metric determines how far one document needs move in order to reach the word embedding of the other.
- Cosine similarity: It calculates the cosine angle between two non-zero vectors to determine their similarity.

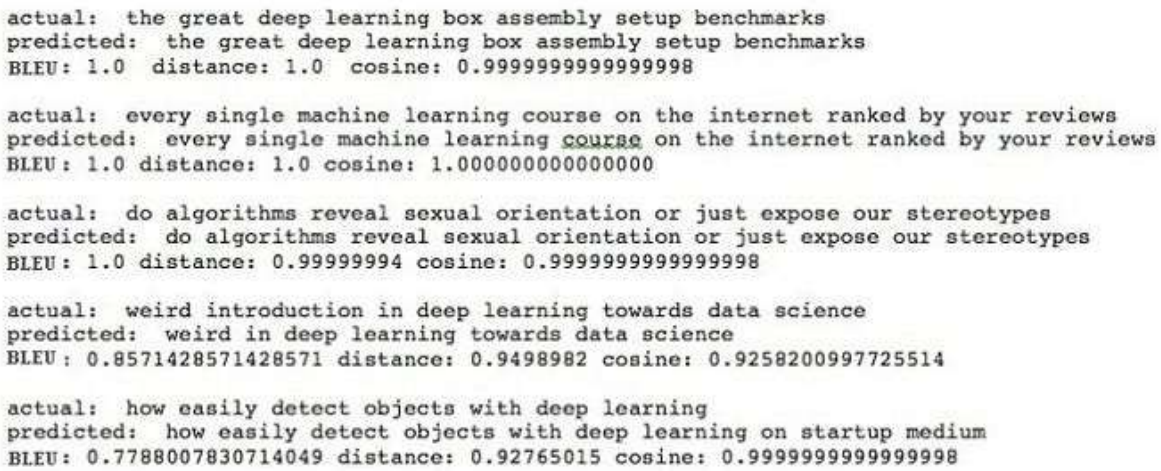

Figure 3: Results of the model on the medium-tech dataset

Some of the key predictions made by our LSTM are shown in the image above. In certain instances, we can observe that the predicted headlines closely resemble the real ones. The bleu scores, cosine similarity, and Google word2vec distance similarity were all calculated after applying the aforementioned model to each individual dataset and the

combined dataset. The complete results are shown down below:

Table 1: comparison table

	Medium tech	BBC business	Law data	NEWS summary
Spacy	84	86	89	91
Vanilla RNN	88	89	91	92
LSTM	89	91	93	90
LSTM with BS	94	96	97	98

The table 1 shows four different dataset (Medium Tech, BBC Business, Law Data, NEWS Summary) are evaluated using four distinct models (Spacy, Vanilla RNN, LSTM, LSTM with BS), with numerical scores representing their performance or accuracy. Across all domains, the "LSTM with BS" model consistently achieves the highest scores, indicating superior performance compared to the other models. The Vanilla RNN and LSTM models also perform well, although with slightly lower scores, while Spacy consistently scores the lowest, suggesting it may be less effective for these specific tasks. These results suggest that the "LSTM with BS" model is the most accurate or efficient choice among the evaluated models for these technology and business-related domains.

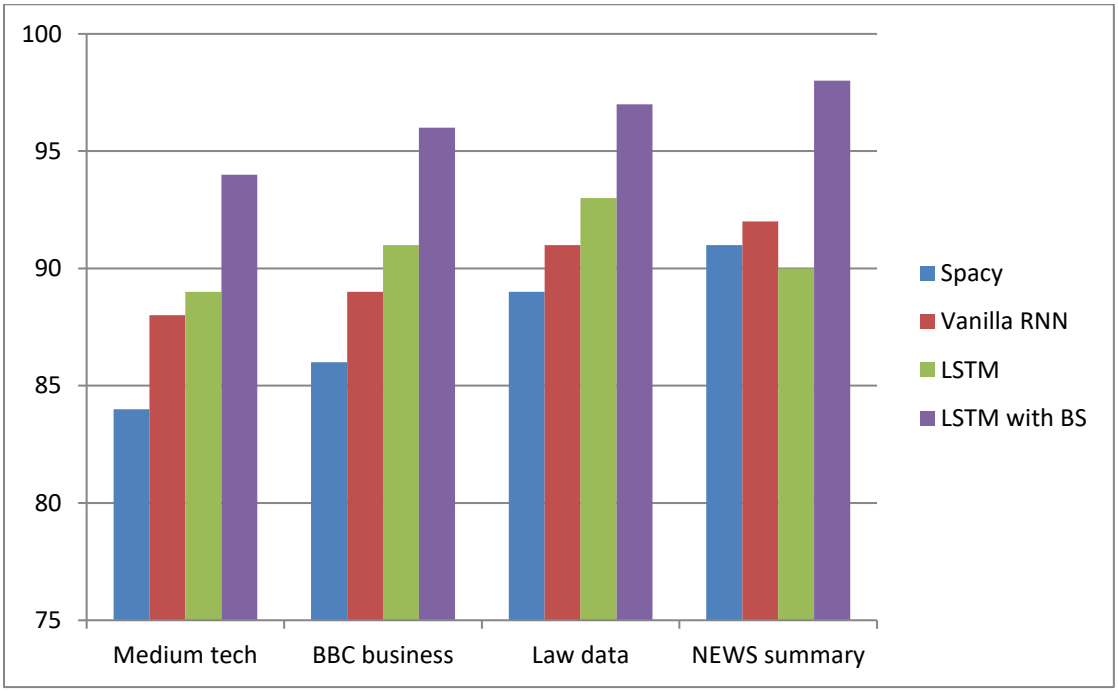

Figure 4: Bleu score of various models of all the mentioned datasets

A spacey-based model, Vanilla RNN as our basic baseline, LSTM, LSTM with beam search, and LSTM with attention were all utilized, with the latter yielding the best results. All of the datasets and models we've used go towards the final calculation for the bleu score. It is evident from these models that LSTM trained with attention achieves superior results.

V. CONCLUSION

The title creation study using several NLP models, including as Spacy, Vanilla RNN, LSTM, and LSTM with Beam Search, has provided helpful insights into the capabilities and limitations of each method. The Spacy model proved its worth in text preparation by extracting crucial information such as named entities and part-of-speech tags, which can then be used to enhance title generation. Spacy, as a stand-alone paradigm, may be incapable of producing unique and contextually relevant titles. Vanilla RNN shown difficulties in capturing long-range associations and retaining memory over longer sequences while operating as a basic recurrent architecture. As a result, its capacity to produce meaningful titles may be limited, especially for more complex content.LSTM, on the other hand, shows a greater ability to capture long-term dependencies, making it more suitable for title generation tasks. Its internal memory and recurrent connections significantly improved the quality of output titles, outperforming Vanilla RNN. The combination of LSTM and Beam Search accelerated the title generation process even more. The combination of LSTM's long-term modeling abilities with Beam Search's ability to evaluate several alternative titles resulted in more diverse and contextually appropriate results. This technique significantly boosted the titles' originality and fluency.

VI. REFERENCE

1. A. Joshi, P. K. Deshmukh and J. Lohokare, "Comparative analysis of Vanilla LSTM and Peephole LSTM for stock market price prediction," 2022 International Conference on Computing, Communication, Security and Intelligent Systems

(IC3SIS), Kochi, India, 2022, pp. 1-6, doi: 10.1109/IC3SIS54991.2022.9885528.

2. C. Liu, L. Zhao, M. Li and L. Yang, "Adaptive Beam Search for Initial Beam Alignment in Millimetre-Wave Communications," in IEEE Transactions on Vehicular Technology, vol. 71, no. 6, pp. 6801-6806, June 2022, doi: 10.1109/TVT.2022.3164533.
3. E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis and K. I. Diamantaras, "Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy," 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece, 2019, pp. 337-341.
4. G. Szűcs and D. Huszti, "Seq2seq Deep Learning Method for Summary Generation by LSTM with Two-way Encoder and Beam Search Decoder," 2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2019, pp. 221-226, doi: 10.1109/SISY47553.2019.9111502.
5. I.C. Irsan, T. Zhang, F. Thung, D. Lo and L. Jiang, "AutoPRTitle: A Tool for Automatic Pull Request Title Generation," 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), Limassol, Cyprus, 2022, pp. 454-458, doi: 10.1109/ICSME55016.2022.00058.
6. K. Kaku, M. Kikuchi, T. Ozono and T. Shintani, "Development of an Extractive Title Generation System Using Titles of Papers of Top Conferences for Intermediate English Students," 2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI), Niigata, Japan, 2021, pp. 59-64, doi: 10.1109/IIAI-AAI53430.2021.00010.
7. K. Liu, G. Yang, X. Chen and C. Yu, "SOTitle: A Transformer-based Post Title Generation Approach for Stack Overflow," 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 2022, pp. 577-588, doi: 10.1109/SANER53432.2022.00075.
8. L. Jain and P. Agrawal, "Sheershak: an Automatic Title Generation Tool for Hindi Short Stories," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 579-584, doi: 10.1109/ICACCCN.2018.8748377.
9. M. Mohseni and H. Faili, "Title Generation and Keyphrase Extraction from Persian Scientific Texts," 2020 25th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 2020, pp. 1-6, doi: 10.1109/CSICC49403.2020.9050113.
10. M. Riku and K. Masaomi, "A title generation method with Transformer for journal articles," 2022 Asia-Pacific Signal and

Information Processing Association Annual Summit and Conference (APSIPA ASC), Chiang Mai, Thailand, 2022, pp. 1115-1120, doi: 10.23919/APSIPAASC55919.2022.9979942.

11. N. Wijaya and D. H. Widyantoro, "Semantic Compositional Network with Top-Down Attention for Product Title Generation from Image," 2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA), Tokoname, Japan, 2020, pp. 1-6, doi: 10.1109/ICAICTA49861.2020.9429031.
12. P. Mishra, C. Diwan, S. Srinivasa and G. Srinivasaraghavan, "Automatic Title Generation for Text with Pre-trained Transformer Language Model," 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2021, pp. 17-24, doi: 10.1109/ICSC50631.2021.00009.
13. T. Cai, C. Wu and J. Zhang, "A Bagging Long Short-term Memory Network for Financial Transmission Rights Forecasting," 2022 7th IEEE Workshop on the Electronic Grid (eGRID), Auckland, New Zealand, 2022, pp. 1-5, doi: 10.1109/eGRID57376.2022.9990015.
14. T. -F. Hsieh, J. -S. Jwo, C. -H. Lee and C. -S. Lin, "Unsupervised Title Generation from Unpaired Data with N-Gram Discriminators," 2022 7th International Conference on Signal and Image Processing (ICSIP), Suzhou, China, 2022, pp. 763-766, doi: 10.1109/ICSIP55141.2022.9886556.
15. T. Zhang, I. C. Irsan, F. Thung, D. Han, D. Lo and L. Jiang, "Automatic Pull Request Title Generation," 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), Limassol, Cyprus, 2022, pp. 71-81, doi: 10.1109/ICSME55016.2022.00015.
16. X. Ma, J. W. Keung, X. Yu, H. Zou, J. Zhang and Y. Li, "AttSum: A Deep Attention-Based Summarization Model for Bug Report Title Generation," in IEEE Transactions on Reliability, doi: 10.1109/TR.2023.3236404.
17. Y. Heng, "A News Title Generation Method Based on UNILM Framework via Adversarial Training," 2021 International Conference on Computer Engineering and Application (ICCEA), Kunming, China, 2021, pp. 37-40, doi: 10.1109/ICCEA53728.2021.00014.