Text Summarization Using Advanced Neural Nets

Ziming Wang
u1431740@utah.edu

Abstract

This research explores the realms of language modeling and text summarization within the scope of Natural Language Processing (NLP), leveraging advanced neural network architectures. The first segment of the study focuses on language modeling using Recurrent Neural Networks (RNNs), with a specific emphasis on bidirectional Long Short-Term Memory (LSTM) networks. Key topics covered include the intricacies of LSTM networks, data preparation and preprocessing strategies, and the challenges faced in language modeling, such as managing sequence lengths and overfitting. The second part of the research pivots to text summarization, utilizing the PEGASUS model, a state-of-the-art transformer-based approach known for its efficacy in generating high-quality summaries. This section outlines the architecture of PEGASUS and fine-tuning strategies tailored for text summarization tasks. Finally, the models are evaluated performance using ROUGE metrics, ensuring a comprehensive understanding of the model's effectiveness in producing coherent and contextually accurate summaries. Together, these studies provide a different view of current advancements in NLP, demonstrating the practical application and evaluation of sophisticated models in language modeling and text summarization.

1. Introduction

In the evolving landscape of Natural Language Processing (NLP), the quest for more sophisticated and accurate language models has become increasingly crucial. This research explores two pivotal aspects of NLP: language modeling, particularly with bidirectional Long Short-Term Memory (LSTM) networks, and text summarization using the PEGASUS model. The urgency of advancing these areas stems from their broad application in technologies like virtual assistants, content summarization tools, and machine translation services. The benefit to end-users, ranging from improved interaction with AI to more efficient information processing, is significant.

Language modeling, the core of NLP, faces challenges in accurately predicting word sequences in large datasets. Traditional methods often fall short in grasping the nuances and complexities of natural language. Bidirectional LSTM networks offer a solution, capable of understanding context from both past and future inputs. However, their optimization for language modeling remains a complex task, often hindered by issues like sequence length management and model overfitting.

Similarly, text summarization is crucial for distilling vast amounts of information into concise, understandable formats. The PEGASUS model, a transformer-based approach, has shown promise in generating high-quality summaries. Yet, fine-tuning and evaluating such models to ensure contextually accurate outputs remains a challenge.

Existing work in these areas, while substantial, does not fully address the complexity and nuances of real-world language processing. This study aims to contribute to this domain by optimizing bidirectional LSTM for enhanced language modeling and fine-tuning PEGASUS for effective text summarization. The purpose is to advance the accuracy and efficiency of NLP models, ultimately benefiting sectors reliant on rapid and accurate language processing and summarization.

Through this research, we aim to address these challenges, providing insights and methodologies that can be leveraged to improve NLP applications in various sectors, thereby enhancing user experience and information accessibility.

2. Related Work/Lit Survey/Background

In exploring advancements in text summarization and language modeling, Wang et al. [1] made significant contributions with their study "Can Syntax Help? Improving an LSTM-based Sentence Compression Model for New Domains." This research delves into enhancing LSTM models for sentence compression, particularly focusing on the role of syntax in improving model adaptability across various domains.

Similarly, the work of Zhang et al. [2] in "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization" represents a landmark in abstractive text summarization. They introduced the PEGASUS model, a pre-training method that focuses on using gap sentences

to generate more coherent and contextually relevant summaries. This approach significantly advances the field of machine-generated text summarization.

3. Approach

Natural Language Processing (NLP) models operate on the principle of predicting the likelihood of a sequence of words. They analyze text data to understand the probability of each word given the words that precede it, essentially learning the patterns and structures of language. By examining the prior context, these models can anticipate the next word in a sequence, as shown in the formula $P(\text{better}|\text{Did you feel})$. This conditional probability is the product of the probabilities of each previous word and its position, reflecting how language models utilize historical information to make informed predictions. The sophistication of these models lies in their ability to handle the complexity and ever-evolving nature of language, adapting to new phrases and usages as they emerge.

First of all, the architecture of a Bidirectional LSTM (BiLSTM) involves two LSTMs: one processes the input data in a forward direction, and the other processes it in a backward direction. This allows the model to capture information from both past (backward) and future (forward) states of the input sequence. Each LSTM cell in a BiLSTM network comprises several components with corresponding equations:

1. Forget Gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
2. Input Gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
3. Cell State: $C_t = f_t * C_{t-1} + i_t * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
4. Output Gate: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
5. Final Output: $h_t = o_t * \tanh(C_t)$

In these equations, σ denotes the sigmoid function, $*$ represents element-wise multiplication, W and b are weights and biases for different gates, h_t is the hidden state at time t, and x_t is the input at time t. The outputs from both the forward and backward LSTMs are typically concatenated at each time step, providing a comprehensive representation of the input data from both directions.

In this project, the BiLSTM network framework starts with an Input Layer, which is configured to handle input sequences of a fixed length. This layer's job is to receive the sequence of tokens to be processed. Next is the Embedding Layer, which has an output dimension of 64. It translates token indices into dense vectors of fixed size and is typically used to process text data, helping the model understand the word context within sequences. Following the embedding is the Bidirectional LSTM Layer, with 200 units indicating the combined number of units for both the forward and backward LSTMs (100 each). This layer processes sequences from both ends, improving the context understanding by providing the network with information from both past and future states. Lastly, there's a Dense Layer with 1641 units, which likely corresponds to the model's output vocabulary. It's the final layer, producing a vector with a length equal to the vocabulary size, and it's usually followed by a softmax activation function to output a probability distribution for the next word in the sequence.

This design leverages the strengths of LSTMs in sequence processing and the bidirectional context to enhance the model's understanding and prediction of sequential data.

Then, the PEGASUS model is for text summarization. The PEGASUS model is built upon the transformer architecture and employs a novel pre-training objective known as "gap sentence generation." In this approach, the model is pre-trained by selecting sentences from a document to form a pseudo-summary and then masking these sentences in the document. The pre-trained model then learns to predict the masked sentences. This strategy simulates summarization because the model must understand the document context and generate coherent, summary-like sentences (Zhang, 2020) [2].

The architecture uses the standard transformer mechanisms with self-attention and feed-forward neural networks. The equation can represent the self-attention mechanism in transformers:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where Q, K, and V are the query, key, and value matrices, and d_k is the dimensionality of the keys.

In my project, I deploy a two-phase approach to text processing. The initial phase, Language Understanding, employs a Bidirectional Long Short-Term Memory (BiLSTM) network to perform a detailed analysis of the raw text. This phase is aimed at empowering the model to discern the contextual nuances and the semantic architecture of the text, allowing it to identify critical entities.

Subsequently, in the Summary Generation phase, the insights gained from the BiLSTM's analysis are channeled into a text summarization model—specifically, the PEGASUS model. Utilizing the contextual and semantic understanding gleaned from the BiLSTM, PEGASUS is tasked with pinpointing the essential elements of the information and crafting a reasonable summary.

This process involves first processing the text through the BiLSTM model to capture the output features or contextual information. These features are then fed into the PEGASUS model as additional context for summary generation. By integrating the linguistic comprehension of the BiLSTM with

the summarization capabilities of PEGASUS, our methodology harnesses the synergistic potential of both models, thereby enhancing the overall text-processing efficacy.

Finally, I examine the accuracy of the last predicted text summary by using ROUGE as a metric for evaluating automatic text summarization and machine translation. It works by comparing automatically generated summaries or translations to a set of reference summaries (usually manually generated, in this case, the highlights column in the dataset). ROUGE is particularly useful because it captures content overlap between automatic summaries and reference summaries. There are many variations of ROUGE, but in this project I used the following three (Lin, 2004) [3]:

1. ROUGE-L: Focuses on the longest common subsequence (LCS) between the system-generated summary and the reference summaries. It does not require consecutive matches but in-sequence matches that reflect sentence-level word order.

2. ROUGE-1: This evaluates the overlap of unigrams (individual words) between the machine-generated summary and the reference summaries. It is a measure of the exact match of words, reflecting the content overlap.

3. ROUGE-2: This assesses the overlap of bigrams (pairs of consecutive words) between the machine-generated summary and the reference summaries. It captures the co-occurrence of two consecutive words, providing insight into the phrasal structure of the content.

4. For ROUGE-1 and ROUGE-2, the formulas for precision (P), recall (R), and F1-score (F) are:

$$P = \frac{\sum_{s \in \text{SystemSummaries}} \sum_{gram_n \in s} \text{Count}_{\text{match}}(gram_n)}{\sum_{s \in \text{SystemSummaries}} \sum_{gram_n \in s} \text{Count}(gram_n)}$$

$$R = \frac{\sum_{r \in \text{ReferenceSummaries}} \sum_{gram_n \in r} \text{Count}_{\text{match}}(gram_n)}{\sum_{r \in \text{ReferenceSummaries}} \sum_{gram_n \in r} \text{Count}(gram_n)}$$

$$F = \frac{(1 + \beta^2) \cdot P \cdot R}{(\beta^2 \cdot P) + R}$$

where $\text{Count}_{\text{match}}(gram_n)$ is the count of n-grams in both the system and reference summaries, and β is typically set to 1 for the F1-score, which equally weights precision and recall.

4. Experiments

This section begins with what kind of experiments you're doing, what kind of dataset(s) you're using, and what is the way you measure or evaluate your results (evaluation metrics). It then shows in detail the results of your experiments. By details, we mean both quantitative evaluations (show numbers, figures, tables, etc.) as well as qualitative results (show images, example results, etc.). Highlight the key takeaways from your experiments.

	article	highlights	id
0	Posh students at Scotland's top . university h...	Boys from St .\nAndrews University drench them...	c71e6c39ad4ac1a6d288e826ace3ab3f1b209a81
1	By . Chris Greenwood . and Eleanor Harding . a...	Detectives believe Meena Patel, 54, target of ...	e5a3364ef1cf5e34b071658bee00636c01469a07
2	Claims by a former senior Victorian government...	Don Coulson, an adviser to former premier Ted ...	81d518f7d5a8b614c0012c7590669fadd2f72e5b
3	A frustrated Michael Moore, the liberal filmma...	Angry left-wing filmmaker trashed Democratic p...	56a4694b6c704acb9a940640d8096e051581 6dbc
4	(CNN) -- This spring break, thousands of colle...	"Alternative" spring breaks are becoming more ...	33dd97439a3b1ede5bc4f24aaf71a411e9ae76c9
...	...	...	...
995	(CNN) -- With Pope Benedict XVI leaving the pa...	Benedict changed a rule, which means the selec...	a4e6ee2b50a0934cf4e2f4f296dce0b297b731da
996	By . Sarah Griffiths . PUBLISHED:. 08:51 EST,...	The Ps Postcards app allows a user to upload t...	28fd1c2d251e88034ae2c239e3f458f92b4760a3
997	Police today continued to scour the banks of a...	Alice Gross, 14, was last seen walking near we...	b48ece171487f5d894986a31a635082552d58d41
998	By . Sarah Griffiths . PUBLISHED:. 12:29 EST,...	The Swedish-designed helmet has a system of st...	8eb9ceb533b7922f65e548d1afef361f9ccd19ae
999	Johannesburg, South Africa (CNN) -- South Afri...	Jackie Selebi, the country's former national p...	0d90e5cbbaae9b35f538e1e83dc709b1080da32f

Figure 1. CNN/Daily Mail dataset

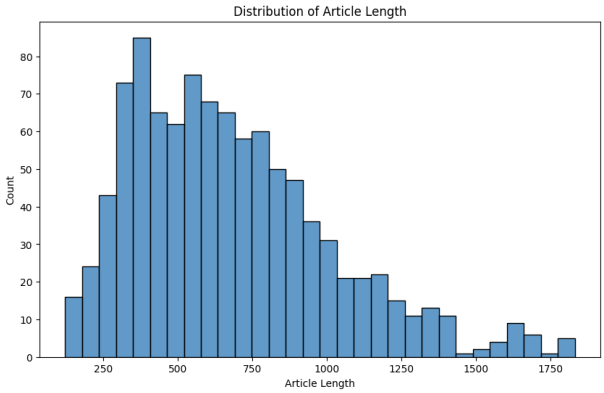

Figure 2. Distribution of Article Length

The dataset I am using is the CNN/Daily Mail dataset. This is a widely used dataset for natural language processing tasks such as text summarization. This dataset contains news articles from CNN and the Daily Mail, along with associated highlights, which act as summaries. The dataset contains more than 300,000 news articles. The article itself is detailed and often lengthy, including citations, background information, and statistics, while the abstract is concise, typically containing 3-4 bullet points that distill the essence of the article. These articles serve as the input of the summary model, and the key points serve as the target output to guide the model to learn the summary task. The data is usually divided into training set, validation set and test set. Summarization models trained on this dataset are evaluated using the ROUGE metric, which measures the quality of summaries by how much they overlap with human-written reference summaries.

The first thing we need to do is to preprocess the dataset, which is a critical step in NLP tasks and varies greatly depending on the goal. In the CNN/Daily Mail dataset, preprocessing consists of cleaning up the text by removing redundant lines or spaces that may introduce noise to the data. Non-alphabetic, non-numeric, or non-punctuation charac-

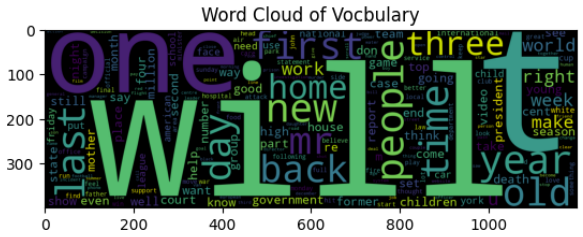

Figure 3. Word Cloud

	count
count	33314.000000
mean	21.140992
std	330.747971
min	1.000000
25%	1.000000
50%	2.000000
75%	7.000000
max	39640.000000

Figure 4. Dataset Words Count

	word	count
0	the	39640
1	to	19129
2	a	17138
3	and	16202
4	of	15929
...	...	...
2704	chain	30
2705	ian	30
2706	solution	30
2707	pointed	30
2708	rory	30

Figure 5. Dataset Words After Masking

ters are also removed to maintain a consistent set of tokens for the model to learn. I then use tokenization to break down the text into these consistent units, which is critical for the model to effectively process and learn from the input data. Tokenization breaks down the text into smaller, more manageable units (called tokens). This process reduces complex text strings to structured elements, creating a vocabulary that machines can interpret and learn from. Tokenization allows the model to process token sequences to understand each token's probability and context relative to its predecessors. In addition, tokenization ensures that the input data meets the expectations of typically pre-trained models that often need to tokenize text in a specific way for optimal performance. Finally, I add "start" and "end" tags to each sequence to enable the model to recognize the beginning and end of each sentence. This step is particularly useful in tasks such as summarization. This step is particularly useful in tasks such as summarizing. It facilitates model learning by providing clear structural markers for text sequences.

After completing the preprocessing of the data, especially after tokenization. This dataset can be more easily counted by the statistics of the words in the counted text. Among them, I found that 75 percent of the words appeared rarely or maybe even only a few times. So the next thing I want to do is to limit the vocabulary to words that occur frequently in the corpus. First of all, it helps to reduce the dimensionality of the data so that machine learning models can be trained more efficiently since less frequent words usually generate noise rather than informative signals. By focusing on words that occur at least 30 times, we can ensure that the model is trained on words with sufficient contextual examples, thus enhancing the model's ability to learn meaningful patterns and relationships. This threshold helps filter out misspellings, rare words, or domain-specific terms that require a disproportionately large amount of data to properly understand their context. In addition, it improves the generalization ability of the model as it is trained on more representative data of the language used in the corpus. I need NLP models that generalize better rather than models that are more prone to overfitting. Therefore in this step, I restrict words that do not occur frequently in the corpus.

After removing a few more words that didn't appear as often, I realized that each sentence was now a different length. So the next thing I'm going to do is standardize the input by padding and truncating standardized sentence lengths. Performing an interquartile range (IQR) analysis here to determine the ideal sentence length ensures that the majority of the data representation will not be affected by outliers. Sentences that exceed the maximum length are truncated to remove less important information, which is usually located at the beginning or end of the sentence, while shorter sentences are usually padded with special markers to meet the desired length. This consistency is critical because many machine learning models expect inputs to have the same dimensions. Standardized lengths facilitate batch processing, which improves computational efficiency and model performance. By choosing the right sentence length based on IQR analysis, the model can retain the most important content without allocating unnecessary resources to rare long sentences. After performing an IQR analysis, I chose to set the maximum length of a single sentence to 60. This allows for a large number of sentences to be taken into account while eliminating as much useless information as possible.

At this point, I have completed most of the dataset preprocessing phase. Before proceeding with bidirectional LSTM modeling, I need to use cumulative sentence generation for language modeling. This is used to construct sequences of words step by step to predict subsequent words. This technique is useful for me to train the language model

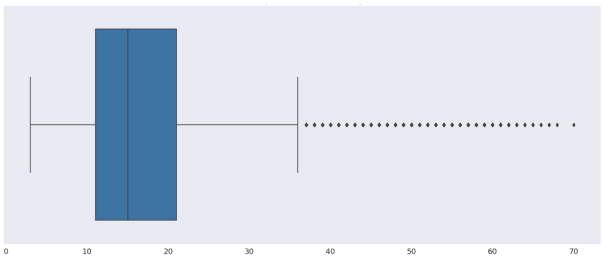

Figure 6. IQR of Sentence Length

	input	output
0	startseq	Order
1	startseq Order	well
2	startseq Order well	received.
3	startseq Order well received.	Items
4	startseq Order well received. Items	nicely
5	startseq Order well received. Items nicely	packed.
6	startseq Order well received. Items nicely pac...	Delivery
7	startseq Order well received. Items nicely pac...	is
8	startseq Order well received. Items nicely pac...	fast.
9	startseq Order well received. Items nicely pac...	Items
10	startseq Order well received. Items nicely pac...	working
11	startseq Order well received. Items nicely pac...	well
12	startseq Order well received. Items nicely pac...	upon
13	startseq Order well received. Items nicely pac...	testing
14	startseq Order well received. Items nicely pac...	endseq

Figure 7. Cumulative Sentence Generation

later on, as it very much mimics the way humans construct sentences, taking into account not only the immediate context but also the broader sentence structure. The advantage of this approach is that it trains the model to predict what comes next in a sentence in a highly contextual and sequential manner. Each step involves the model considering all previous words, thus maintaining the semantic integrity of the sentence. This is essential for generating text that is not only grammatically correct but also contextually coherent. By implementing this strategy, we enable the model to understand linguistic patterns in the text. It is able to better prepare for the text summarization that follows.

I have prepared the inputs for a bi-directional LSTM where I have constructed a bi-directional LSTM neural network. The next layer is the Embedding Layer, whose key role is to convert these strings of words into dense 64-dimensional vectors that capture not only the features of the words, but also the semantics, thus effectively mapping the input into a higher dimensional space. After embedding, a bi-directional LSTM layer with 200 cells processes the sequence forward and backward. This bi-directionality allows

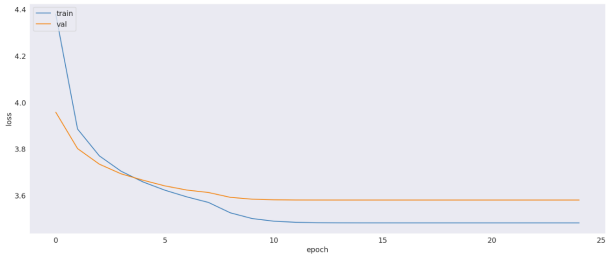

Figure 8. Loss Plot

the model to capture contextual information from past and future data points in the sequence, improving its prediction accuracy by leveraging a more comprehensive understanding of the input text, consistent with the size of the vocabulary used in the model. The purpose of this layer is to output the probability distribution of the next word in the sequence.

According to the loss plot, the training loss drops sharply at the beginning and then continues to drop at a slower rate. But the validation loss is always lower than the training loss, which means that it does not learn the training data too specifically and generalizes well to new data. But there are no obvious signs of overfitting, as the validation loss does not increase as training progresses. Overall, the loss plots show that the model has a good balance between learning from the training data and generalizing to new data.

The next step is to use the bi-directional LSTM output as input to the PEGASUS model.The PEGASUS structure is optimized specifically for the text summarization task, taking advantage of the converter's ability to understand context and generate relevant and concise summaries. The use of bi-directional LSTM outputs as inputs allows the PEGASUS model to benefit from the rich contextual representation captured by the LSTM network, potentially improving the quality of the generated summaries.

The ROUGE-1 score of 0.3192 indicates that approximately one-third of the unigrams in the generated summaries match those in the reference summaries, which points to an adequate grasp of the key terms within the texts. However, the bigram matching score, ROUGE-2, is 0.1257, revealing that the model struggles to construct bi-gram phrases that align with the reference texts. This could signal that while the model is picking up individual keywords, it may not be stringing them together in the same way humans do. The ROUGE-L score, which reflects the longest common subsequence, is closer to the unigram score at 0.2928, indicating better performance in maintaining sentence structure from the reference summaries.

5. Conclusions

Overall, this project provides valuable insights into the complexities of language modeling and summarization.

	id	article	highlights	predictions
0	92c514c913c0bdfe25341af9fd72b29db544099b	Ever noticed how plane seats appear to be gett...	Experts question if packed out planes are put...	While United Airlines has 30 inches of space, ...
1	2003841c7dc0e7c5b1a248f9cd536d727f27a45a	A drunk teenage boy had to be rescued by secur...	Drunk teenage boy climbed into lion enclosure ...	Next level drunk: Intoxicated Rahul Kumar, 17,...
2	91b7d2311527f5c2b63a65ca98d21d9c92485149	Dougie Freedman is on the verge of agreeing a ...	Nottingham Forest are close to extending Dougi...	Freedman has stabilised Forest since he replac...
3	caabf9cbdf96eb1410295a673e953d304391bfbb	Liverpool target Neto is also wanted by PSG an...	Fiorentina goalkeeper Neto has been linked wit...	Liverpool target Neto is also wanted by PSG an...
4	3da746a7d9afcaa659088c8366ef6347fe6b53ea	Bruce Jenner will break his silence in a two-h...	Tell-all interview with the reality TV star, 6...	Speaking out: Bruce Jenner, pictured on 'Keepi...
...	...	...	...	...

Figure 9. PEGASUS Predictions

The use of bi-directional LSTMs shows promise for capturing subtle linguistic patterns, while integration with PEGASUS further improves the ability to generate concise summaries. The moderate ROUGE scores obtained indicate success in the basics, but also highlight the need for further improvements. Future work could explore enhancing the dataset. Beyond this, to improve the performance of the PEGASUS model, we could consider incorporating more contextual training, fine-tuning it using a wider range of datasets, or exploring advanced architectural features that can better capture the sequential and hierarchical nature of language. The bidirectional LSTM also showed some anomalous results. This could be improved in the future with more tuning for bidirectional LSTM or using better techniques.

References

[1] Liangguo Wang, Jing Jiang, Hai Leong Chieu, Chen Hui Ong, Dandan Song, and Lejian Liao. Can syntax help? improving an LSTM-based sentence compression model for new domains. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1385–1393, Vancouver, Canada, July 2017. Association for Computational Linguistics. 1

[2] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2020. 1, 2

[3] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. 3