Cardiff University
School of Computer Science and Informatics

CM3203 – One Semester Individual Project

Title Generation from Abstract Using Deep Learning Techniques

Supervisor: Dr Xianfang Sun

Author: David Priehoda

19/05/2023

Abstract

The task of generating informative and concise titles for academic papers presents a challenging problem, necessitating the ability to understand abstract content and condense it effectively. This project aims to develop a software system that automates title generation from paper abstracts using natural language processing and deep learning techniques. I focus on implementing and comparing the performance of two deep learning models: a bidirectional LSTM with attention and a transformer model known as T5. In addition, I will discuss the process of gathering, preprocessing, and normalising the data to ensure compatibility with the deep learning algorithms. The evaluation of both models will identify the most accurate and relevant technique for generating titles from abstracts.

Table of Contents

Introduction

As the global publication output experiences continuous growth each year[1], the importance of well-crafted titles is becoming increasingly significant. The title of a paper serves as the first point of contact between the research and its potential readers, making it a crucial element in capturing attention and conveying the essence of the work. Generating an informative and concise title can be a time-consuming and challenging task, especially when researchers need to balance content clarity and brevity. A title generation system can alleviate this burden, offering an efficient and consistent method for producing effective titles.

The intended audience of such a system includes, but is not limited to, researchers, academic institutions, and publishers. Researchers will be able to save time and effort in the title creation process, focusing their energy on the content of their work. Academic institutions and publishers can streamline the review and publication processes, as concise and informative titles make it easier to assess the relevance and quality of submitted research papers. Ultimately, a title generation system will benefit the entire scientific community by improving the visibility and accessibility of published research.

This main objectives of this project are: (1) Implementation of two state-of-the-art deep learning models, namely the bidirectional LSTM model with attention and the T5 transformer model, for generating titles from abstracts. (2) Optimisation of both models through fine-tuning and experimentation. (3) Comparison of the performance of the two models to identify the most effective technique for title generation. (4) Development of a software system that can generate informative and concise titles based on abstracts, providing a practical solution for the scientific community.

After rigorous experimentation and evaluation, it is evident that the T5 Transformer model significantly outperforms the bidirectional LSTM with attention in generating titles from abstracts. The developed software system, powered by the T5 model, provides a practical solution for generating informative and concise titles, streamlining the workflow for researchers, academic institutions, and publishers alike.

Background

Before I was able to make a start with designing the deep learning models, I had to research the techniques and the software libraries which would make this project possible. This proved to be quite a learning curve for me as I had no prior experience with machine learning. Some of the things I researched are: natural language processing, RNNs & LSTMs, Attention, Transformer models, evaluation metrics for NLP, tools and libraries for deep learning.

Overview of machine learning

Machine learning (ML) is a subfield of artificial intelligence that enables computers to learn from and make predictions or decisions based on data. It was defined as “the field of study that gives computers the ability to learn without explicitly being programmed” by AI pioneer Arthur Samuel in the 1950s. Machine learning algorithms are first trained on a set of input and target output data during which the algorithm is looking for patterns. After training the algorithm can be fed some unseen input data and it will use the patterns it has previously learned to make a prediction on what the output should look like. [2]

While ML has received massive amounts of attention in recent years, it is not at all a new concept. Rather, research into machine learning started in the fifties, when early computer scientists and mathematicians began exploring algorithms and models for artificial intelligence. Pioneers like Arthur Samuel, Frank Rosenblatt and Marvin Minsky laid the foundation for this rapidly evolving field. Over the decades, advancements in computational power, data availability, and algorithmic innovation have accelerated the development and widespread adoption of ML techniques. Today, machine learning has transformed from an obscure academic pursuit into a powerful and indispensable tool that is revolutionising industries and reshaping the way we live and work.

For this project, I will be using deep learning, which is a subset of machine learning that focuses on networks with many hidden layers. These layers enable the network to capture complex relations and extract intricate patterns from large sets of unstructured data such as images, text and audio. Deep learning has become the driving force behind the cutting edge applications being developed today such as natural language processing (which I will be focusing on in this project), speech recognition and computer vision.

Natural language processing

Natural language processing (NLP) is a area of artificial intelligence which focuses on enabling computers to understand, interpret and generate human language in a meaningful way [2]. For this project I will use natural language processing techniques for preprocessing the the input data such as removing stop words, removing punctuation and tokenisation.

Firstly, stop words are words which can be removed from the text as they do not add much meaning; “the”, “is” and “and” are a few examples. Removing these words helps with reducing the size of the dataset, which in turn helps reduce computational complexity.

Tokenisation is a another preprocessing step needed to train a deep learning model on text data. For my project I will be using word level tokenisation which is assigning each word in the dataset a unique token. A token is simply a number, for example if the token 1 is assigned to the word “hello” and 2 is assigned to “world” then the phrase “hello world” can be represented as 1,2. Tokenisation helps with converting the unstructured text data into a structured format wherein each word is a number and allows for text data to be represented as a vector which allows a deep learning model to look for patterns between input and output vectors.

Recurrent Neural Networks (RNNs)

RNNs are a class of artificial neural networks designed to handle sequential data. Unlike traditional feed forward neural networks, RNNs maintain an internal state that can capture information from previous steps in the sequence, making them suitable for tasks involving sequences, such as natural language processing. RNNs are capable of learning patterns from paper abstracts and corresponding titles and then generating output titles when given an unseen paper abstract.

An RNN is structured with input, hidden, and output layers, but unlike feed forward networks, it has recurrent connections that allow information to be passed along the sequence. At each time step, the RNN receives the current input along with the previous hidden state. The hidden state is updated based on this input and passed to the next time step, forming a loop that enables the RNN to maintain memory of past inputs.

The equation for a hidden state at time t can be represented as:

$$\mathrm{h(t)} = \mathrm{f}(\mathrm{W}_{hh} h_{t-1} + W_{xh} x_t + b_h)$$

- Where h(t) represents the hidden state at time step t.
- f is the activation function, which can be the hyperbolic tangent (tanh) or another suitable function, such as the sigmoid or ReLU function.
- W_{hh} is the weight matrix for the hidden-to-hidden connections, controlling the contribution of the previous hidden state h_{t-1} to the current hidden state h(t).
- h_{t-1} represents the hidden state at the previous time step.
- W_{xh} is the weight matrix for the input-to-hidden connections, controlling the contribution of the current input X_t to the hidden state h(t).
- X_t is the input at the current time step t.
- b_h is a bias term for the hidden layer, which is used to adjust the output of the activation function and improve the flexibility of the model.

The equation for the output at each time step can be represented as:

$$\mathrm{y}_t = W_{hy} h_t + b_y$$

- Where y_t represents the output at time t
- W_{hv} represents the weight matrix for the hidden-to-output connections, controlling the contribution of the hidden state h_t to the output y_t
- h_t is the hidden state at the current time step t
- b_y is the bias term for the output layer, which is used to adjust the output value

Long Short Term Memory (LSTM)

While RNNs are well-suited for sequential data, they suffer from the vanishing gradient problem, which makes it difficult for them to learn long-range dependencies in the input sequences. This limitation can be addressed by using a more advanced architecture, such as LSTM networks, which are specifically designed to tackle this issue.[7]

LSTM networks are a type of RNN that use specialised memory cells in place of the standard hidden state nodes found in basic RNNs. These memory cells are designed to store information over longer sequences, thereby mitigating the vanishing gradient problem and enabling the network to learn longer dependencies. The LSTM memory cell comprises three gates: the input gate, the forget gate, and the output gate, which work together to control the flow of information into, within, and out of the cell.

The equations governing a LSTM cell can be represented as follows:

Input gate:

$$\mathrm{i}_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

Forget gate:

$$\mathrm{f}_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

Output gate:

$$\mathrm{o}_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

Where:

x_t is the input at time t

h_{t-1} is the hidden state at the previous time step

i_t, f_t, o_t are the input,forget and output gates respectively at time t

$W_{ix}, W_{ih}, W_{fx}, W_{fh}, W_{ox}, W_{oh}$ are the learnable weights of the LSTM cell

b_i, b_f, b_o are learnable bias vectors

LSTM networks have been widely adopted in numerous NLP tasks, thanks to their ability to effectively model long sequences of text. For this project, LSTM networks can be used to learn complex patterns within the abstracts of papers and generate relevant titles accordingly. By incorporating an attention mechanisms, LSTMs can focus on the most salient parts of the input data, further enhancing the quality of the generated titles.

Attention

Attention was created to improve the performance of deep learning models, especially in sequence-to-sequence tasks such as translation or summarisation. A attention mechanism allows the model to selectively chose parts of the input sequence to focus on while ignoring other, less significant, parts when generating the output. This is particularly useful when the input sequences are long. By implementing attention the model can capture long-range dependencies more effectively in the input sequence which results in improved results.

The fundamental concept behind the attention mechanism is to compute a context vector, which is a weighted sum of the input hidden states where the weights are determined by the attention weights. The attention weights represent the significance of each input hidden state for generating the current output and is calculated by a function which measures the relevance of each input hidden state to the current output hidden state.

Here is how the attention mechanism works:

1. Compute the attention scores for each input hidden state:

$$e_{tj} = a(h_t, s_{j-1})$$

The attention score is a measure of the similarity between the input hidden state h_t and the output hidden state s_{j-1}. The attention score is calculated using a function a, which can be any function that measures similarity, a common choice is dot product

2. Calculate the attention weights

$$\alpha_{tj} = \mathrm{softmax}(e_{tj})$$

The attention weight is a normalised version of the attention score. The attention weight is calculated using a softmax function, which ensures that the attention weights sum to 1. This means that the attention weights can be used to create a weighted sum of hidden states.

3. Compute the context vector

$$c_j = \sum_t \alpha_{tj} h_t$$

The context vector is a weighted sum of the input hidden states. The weights are determined by the attention weights.

4. Generate the output hidden state

$$s_j = f(c_j, s_{j-1}, y_{j-1})$$

The output hidden state is the hidden state of the decoder at time step j. The output hidden state is calculated using a function f, which takes the context vector, the previous output hidden state, and the previous output as input.

Transformers

Transformers are a type of deep learning architecture introduced by Vaswani et al. in their 2017 paper titled ‘Attention Is All You Need’, which has revolutionised NLP and achieved unprecedented performance in tasks such as machine translation, summarisation and title generation.[6]

The main components of the architecture are a self-attention mechanism, positional encoding, encoder and decoder layers and multi-head attention. With the main innovations being the self-attention mechanism and positional encoding, which enable the model to process input sequences in parallel and effectively capture long-range dependencies.

Self-Attention Mechanism

The self-attention mechanism enables the model to weigh the importance of each token relative to other tokens in the input sequence. It is similar to the attention mechanism described in the previous section and also calculates attention scores, weights and context vectors however it does so for each token in the input sequence.

Positional Encoding

As the transformer processes input sequences in parallel, it lacks the concept of token order. This is addressed by positional encodings which are added to the input embeddings. They provide information about the position of each token in the sequence.

Encoder and Decoder layers

The encoder layers are responsible for processing the input sequence. Each encoder layer consists of multi-head self-attention, followed by a feed-forward network. The multi-head self-attention mechanism allows the encoder to attend to different parts of the input sequence at the same time. This allows the encoder to learn long-range dependencies in the input sequence. The feed-forward network allows the encoder to learn more complex features from the input sequence.

The decoder layers are responsible for generating the output sequence. Each decoder layer consists of an attention layer, followed by a feed-forward network. The attention layer allows the decoder to attend to both the input sequence and the output sequence that it has generated so far. This allows the decoder to generate output that is both relevant to the input sequence and grammatically correct.

The Encoder and Decoder layers are repeated multiple times to form the model with the number of layers being a hyper parameter that can be tuned accordingly.

Multi-head Attention

Multi-head attention is an attention mechanism which allows the model to attend to different parts of the input at once by splitting the input sequence into multiple “heads” and then attending to each head independently. The outputs of each attention head are then concatenated to produce the final output.

Transfer learning

Transfer learning is a technique which enables the fine tuning of pre-trained models to improve performance and reduce the training time needed to learn a new task. The main idea behind it is that the knowledge learned by a model for one problem can be used as a starting point for another similar problem thus reducing the amount of data needed and the training time.

In the context of NLP, large language models such as GPT, BERT and T5 have already been trained on massive datasets of text and have very good performance in generating text. These can be fine tuned for a specific NLP task such as summarisation by using transfer learning, resulting in improved performance and faster convergence.

Evaluation metrics

ROUGE

ROUGE is a set of evaluation metrics which measures the quality of generated text by comparing it to a reference text. ROUGE focuses on the recall of n-grams (sequences of n words) and computes three scores: ROUGE-N, ROUGE-L and ROUGE-S.

Firstly, ROUGE-N calculates the n-gram recall between generated text and reference text. It is calculated for different numbers of n such as ROUGE-1, ROUGE-2 and so on. A higher ROUGE-N score indicates a better match of n-grams in the texts.

$$ROUGE - N = \frac{|S_n \cap G_n|}{|S_n|}$$

Where:

S_n is the set of all n-grams in the reference text

G_n is the set of n-grams in the generated text

Secondly, ROUGE-L measures the longest common sequence of words between the generated and reference text. The longest common sequence takes into account both the word order and the presence of words in both texts.

$$ROUGE - L = L_{lcs}/L_{max}$$

Where:

L_{lcs} is the length of the longest common sequence of words between the generated and reference text

L_{max} is the length of the longer of the two texts

Lastly, ROUGE-S calculates the skip-bigram recall between the generated and reference texts. Skip-bigrams are pairs of words in a sentence, allowing for gaps between them. By considering skip-bigrams, ROUGE-S captures more flexible and diverse patterns in word arrangements, making it more robust against variations in word order.

BLEU

BLEU is another popular evaluation metric for NLP tasks. It also measures the similarity between generated and reference text by comparing n-gram overlap. BLEU is more precision oriented as it penalises generated text for including words which are not in the reference text. Scores range from 0 to 1 with higher scores indicating better performance.

METEOR

METEOR is an evaluation metric which addresses some of the shortcomings of ROUGE and BLEU by incorporating both precision and recall into its calculations. It also considers the order of words and phrase matches to provide a more comprehensive evaluation.

Software libraries

1. TensorFlow: TensorFlow is an open-source software library developed by Google for numerical computation and machine learning. It provides a flexible platform for defining, training, and deploying machine learning models.[8]
2. NumPy: NumPy is an open-source library which allows for the use of large, multi-dimensional arrays and matrices efficiently. NumPy is essential for scientific computing with Python and provides numerical operations, which are importatnt for implementing machine learning algorithms.[9]
3. NLTK (Natural Language Toolkit): NLTK is a leading platform for building Python programs to work with human language data (text). It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with a suite of text processing libraries for classification, tokenisation, stemming, tagging, parsing, and semantic reasoning. NLTK is widely used in research and industry for natural language processing tasks, including sentiment analysis and text summarisation.[10]
4. Transformers (Hugging Face): The Hugging Face Transformers library is an open-source library that provides state-of-the-art pre-trained models and architectures for natural language processing tasks such as text classification, text generation, question answering, and more. It includes popular transformer-based models like BERT, GPT-2, T5, and RoBERTa, as well as utilities for training, fine-tuning, and deploying these models. Transformers library aims to make cutting-edge NLP techniques accessible to the wider community with a user-friendly API.[11]

These software libraries, among others, play a significant role in the development and implementation of deep learning and natural language processing models, offering powerful tools and resources. Their general-purpose nature allows them to be applied to a wide range of tasks and projects, including title generation from research paper abstracts.

Challenges in title generation

There are many challenges in the task of title generation which need to be addressed to be able to create a good title generator. From my research here are a few of the main challenges:

1. Vocabulary and Language Ambiguity: Title generation requires a comprehensive understanding of the language and its intricacies. The generated titles should accurately represent the main idea of the given text while being grammatically correct and free from ambiguity. Dealing with synonyms, homonyms, and other linguistic variations is a challenge in generating meaningful titles.

2. Capturing the Essence of the Abstract: An effective title should succinctly capture the core message of the research paper abstract. The challenge lies in identifying the most critical and relevant information from the abstract and condensing it into a concise title without losing important context.

3. Handling Long and Complex Sentences: Research paper abstracts often contain long and complex sentences with domain-specific terminology. Generating a coherent and informative title requires understanding the structure and meaning of these sentences, which can be challenging for machine learning models.

4. Diverse Writing Styles and Terminology: Research papers come from various fields, each with its unique writing style, terminology, and conventions. A title generation system should be adaptable to these diverse styles and be able to generate suitable titles for papers from different domains.

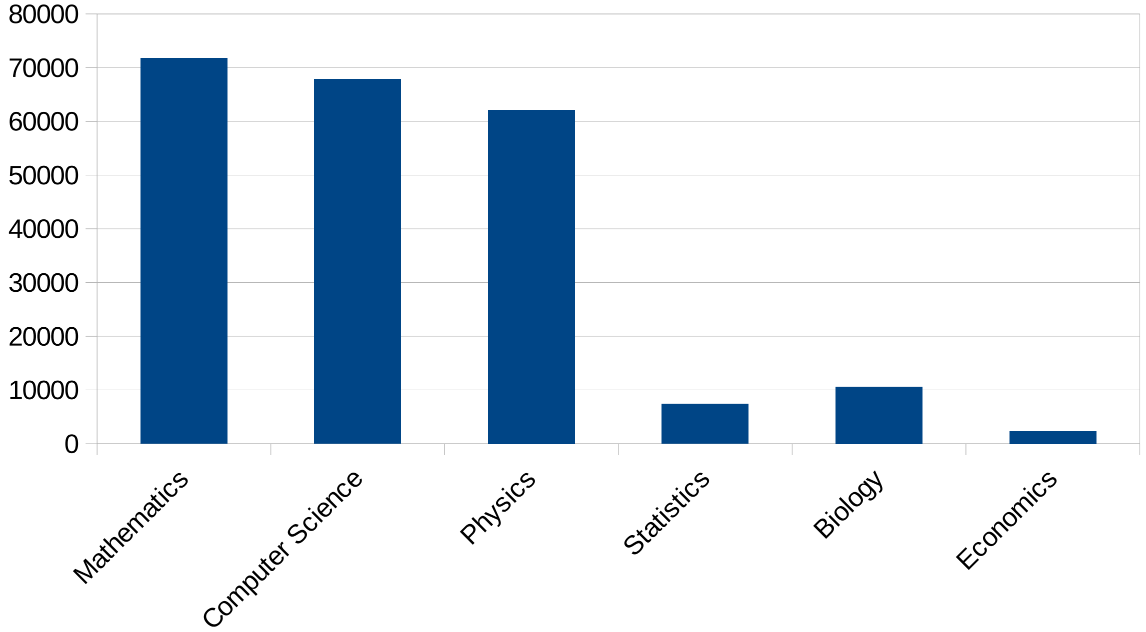
5. Creativity and Novelty: An appealing title should be not only informative but also engaging and creative. Machine learning models can struggle with generating novel and creative titles that can capture the attention of readers while still being informative.

6. Evaluation Metrics Limitations: Evaluating the quality of generated titles can be challenging. Traditional NLP evaluation metrics like ROUGE, BLEU, and METEOR may not fully capture the nuances of title quality, as they primarily focus on n-gram overlap with reference titles. Developing more sophisticated evaluation metrics that consider factors like informativeness, novelty, and creativity is an ongoing challenge in title generation research.

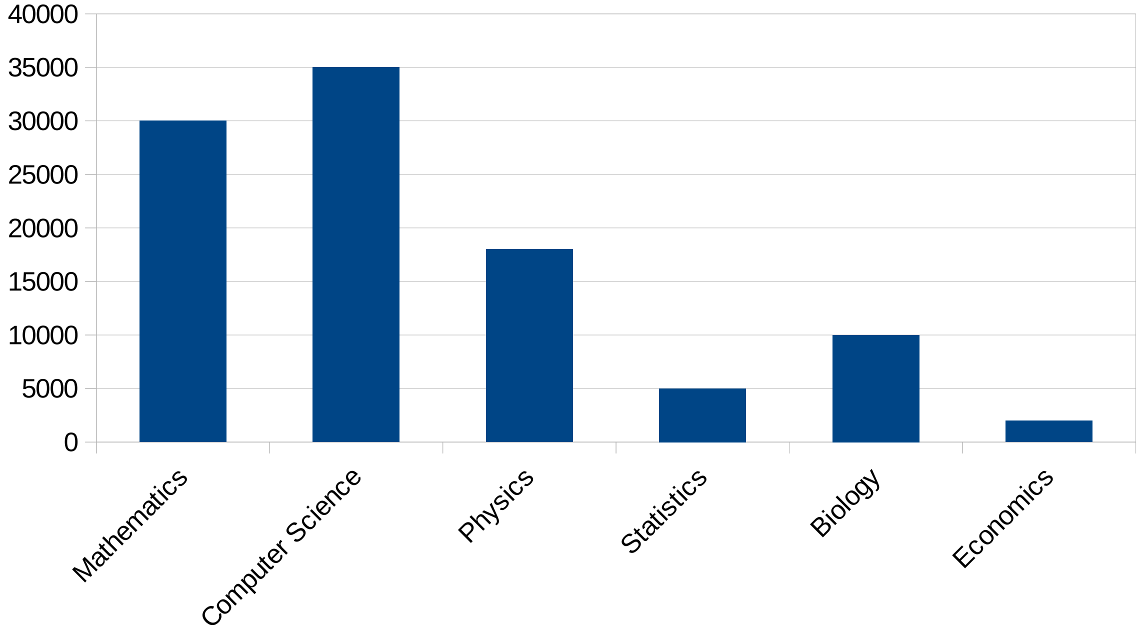
Approach

Data collection and preprocessing

To create a model which can generate titles from paper abstracts I will need a large dataset of titles and their corresponding abstracts to train the model on. To get this I wrote a python script which uses the arXiv API[3] to get the data needed. Using the arXiv API I was able to get a total of 222072 titles and abstracts from the categories of Mathematics, Computer Science, Physics, Statistics, Biology and Economics. The breakdown of how many from each category can be seen in the chart below.



Once I had this data, I created a dataset of 100000 titles and abstracts with the following distribution of categories:



Next I preprocessed and cleaned the dataset by removing punctuation and other special characters, lower casing all text and removing stop words from the abstracts.

Here is a sample of the dataset:

```
                                                Title                                              Abstract Category
0  the galerkin method for perturbed selfadjoint ...  consider galerkin method approximating spectru...  math.SP
1  on a class of nonselfadjoint multidimensional ...  investigate multidimensional schrodinger opera...  math.SP
2  on the failure of fixedpoint theorems for chai...  effective topos exists chaincomplete distribut...  math.CT
3  a study of saturated tensor cone for symmetriz...  let fg symmetrizable kacmoody lie algebra stan...  math.RT
4  stability of a nonlinear axially moving string...  paper nonlinear axially moving string kelvinvo...  math.AP
```

BiLSTM Model

The next step of the project was to implement the first model. I decided to first implement the bidirectional LSTM model with attention.

Data tokenisation

The first step was to preprocess the data in a way that can be used by the model. I created a script which first truncates titles and abstracts to a configurable max length if they are longer than max length and then adds a <start> and <end> token to each title. The script then tokenises the titles and abstracts using separate tokenisers by using the 'Tokenizer' class from the keras library. I added the 'vocab_size_titles' and 'vocab_size_abstract' parameters to define how many unique words each tokeniser should consider. Lastly, the script pads any titles and abstracts which are shorter than the chosen max length with trailing 0s.

Model implementation

After tokenising the data, I moved onto creating the model. This part was more difficult than I was expecting as I ran into multiple issues and errors along the way. The primary objective of the model is to be able to learn patterns between abstracts and titles. To do this I created a class called 'BiLSTMAttention' which combines the power of bidirectional LSTM layers with an attention layer.

The model consists of the following layers:

1. Input layer

   The first layer is an input layer with the shape (self.X_max_len,).

```
inputs = Input(shape=(self.X_max_len,), dtype='int32')
```

2. Masking layer

   The next layer is a masking layer which masks unknown words in the input sequences. This is to make sure the model focuses on the words which are in the vocabulary as unknown words do not provide any useful information.

```
masked_inputs = Masking(mask_value=1)(inputs)
```

3. Embedding layer

Next, I used an embedding layer to convert the tokenised sequences into dense vectors which can then be fed into the model. This layer also masks zeros, which are used as padding.

```
x = Embedding(input_dim=self.X_vocab_len, output_dim=self.hidden_size, mask_zero=True)(masked_inputs)
```

4.BiLSTM layers

The sequences are then passed into one or more bidirectional LSTM layers stacked on top of each other. These are used to capture both the forward and backward context from the sequences enabling the model to learn long range dependencies. I also gave the layers a configurable dropout and recurrent dropout which reduces overfitting by randomly dropping out some of the units in the layer during training, thus preventing them from becoming too dependent on each other and improving generalisation performance.

```
for _ in range(self.num_layers):
    x = Bidirectional(LSTM(self.hidden_size, return_sequences=True, dropout=self.dropout, recurrent_dropout=self.recurrent_dropout))(x)
```

5. Attention layer

After the LSTM layers, I added a Attention mechanism which allows the model to selectively focus on relevant parts of the input sequence.

```
query = x
value = x
attention = Attention()([query, value])
```

6. Output layer

Lastly, the output from the attention layer is passed through to a time distributed dense layer. I used the softmax activation function [4] to produce a probability distribution over the titles vocabulary.

```
outputs = TimeDistributed(Dense(self.y_vocab_len, activation='softmax'))(attention)
```

I then compile the model using the categorical cross entropy loss function and ADAM optimiser.

```
self.model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Inference

Now that I had the model complete, It was time to create an inference function to allow me to generate titles. To do this I created a 'Generator' class which uses a modified beam search algorithm to generate titles from the probability distributions that come from the output layer of the model.

Calling the 'Generator' class takes a abstract as a argument as well as optional arguments for temperature and beam width. The class will first clean and tokenise the input abstract and then pass it to the model and get back a probability distribution over the title vocabulary. The probabilities are then passed to the beam search function along with a temperature and beam width argument which will then decide on which word should come next in the sequence. The beam search function returns a list of tokens which are then decoded back into their corresponding words using the title vocabulary.

Since I was encountering a problem with the model generating very short titles, I modified the beam search so that each time it predicts the end token there is a 50% chance to disregard it and chose the next most likely token. This helped with increasing the length of the titles generated. Additionally, I modified the beam search so that if it chooses the 1 token, which represents unknown words, it will chose the next most likely token instead.

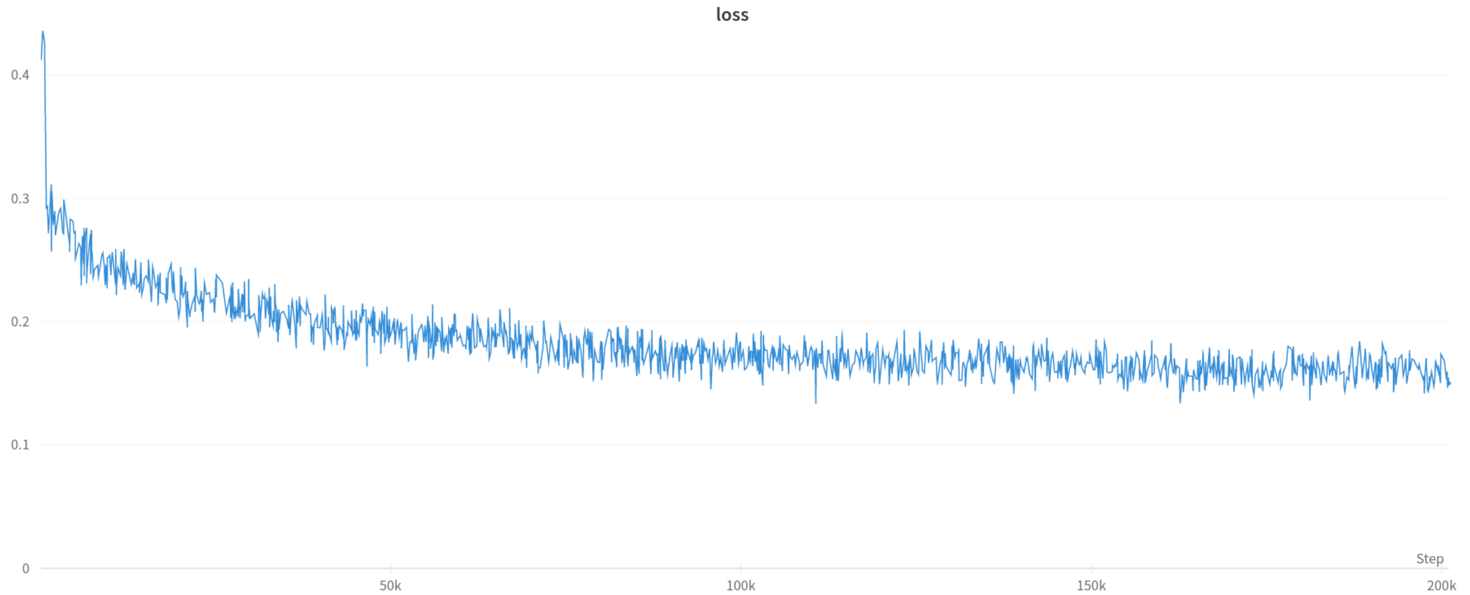
Model training

Next It was time to train some models and find the optimal hyper parameters. This turned out to be very time consuming as each model took a long time to train. Another issue I ran into while training was the RAM and VRAM requirements were quite high. The reason for this is that my model uses one hot encoding across the vocabulary which results in a lot of wasted memory space and limited the vocabulary size and max length that I could use.

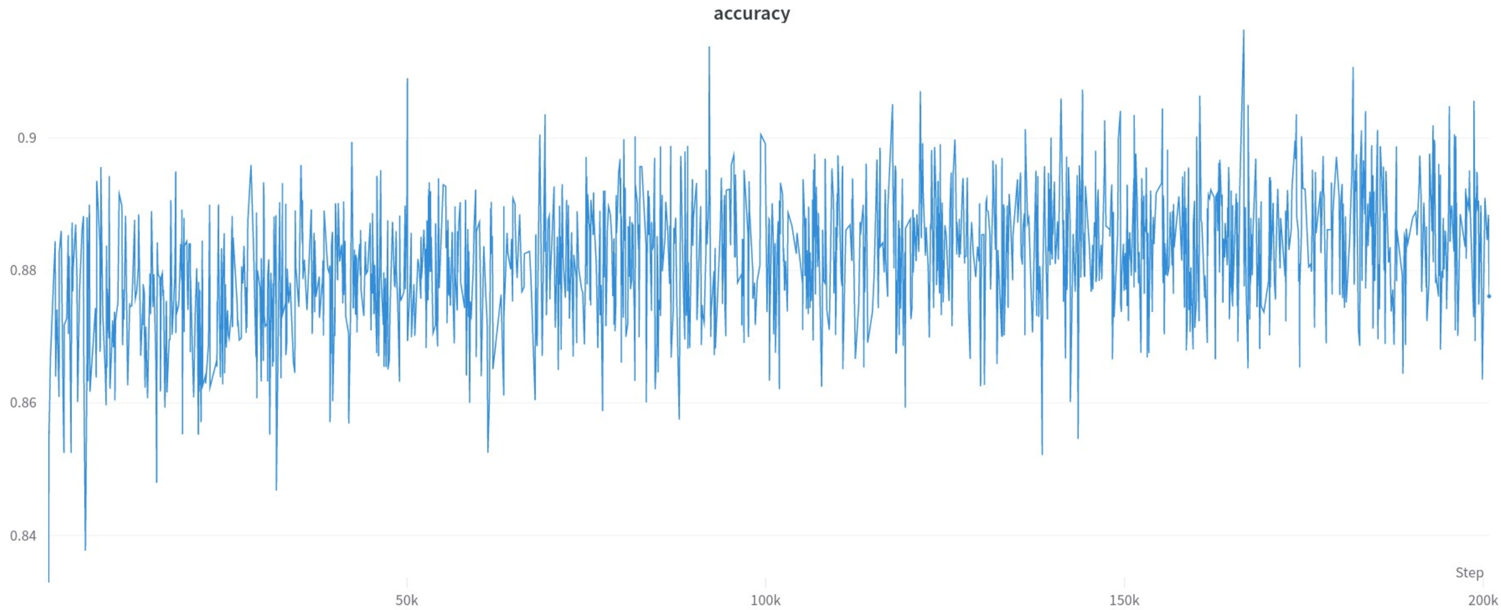
The table below shows the hyper parameters which I used to train several models.

Model	Number of layers	Max Length	Hidden Size	Dropout	Recurrent Dropout	Abstract Vocab Size	Title Vocab Size	Validation Split
1	4	120	128	0.3	0.3	6500	8000	0.2
2	2	60	40	0.8	0.8	6000	5000	0.2
3	2	50	64	0.5	0.5	5000	6000	0.2
4	3	64	32	0.5	0.5	5000	6000	0.2
5	4	120	128	0.6	0.6	6500	8000	0.2
6	2	256	128	0.5	0.5	6500	8000	0.2

After training each for 25 epochs, model number 6 showed the best results in terms of loss and accuracy and so I decided to train that one further. I decided to train it for 200 epochs which took a total of 170 hours and the loss and accuracy graph for it can be seen below.



loss
0.4
0.3
0.2
0.1
0
Step
50k
100k
150k
200k



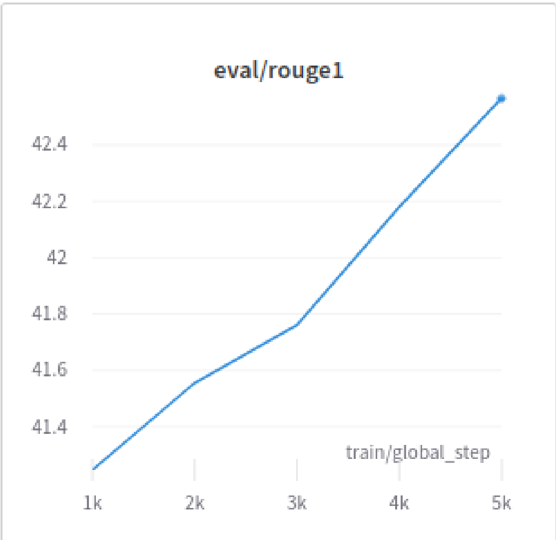
accuracy
0.9
0.88
0.86
0.84
Step
50k
100k
150k
200k

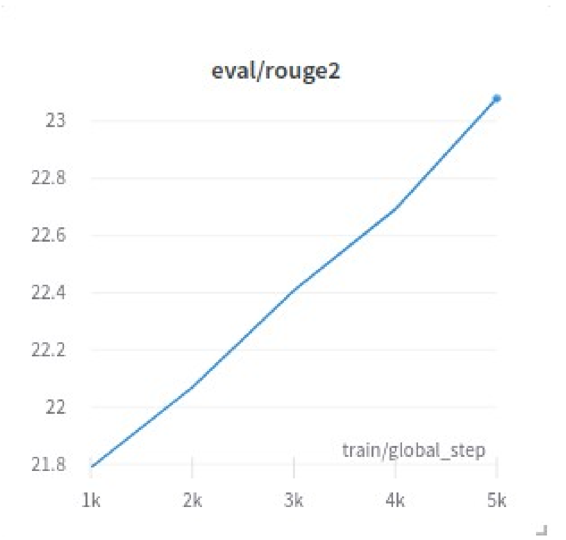
T5

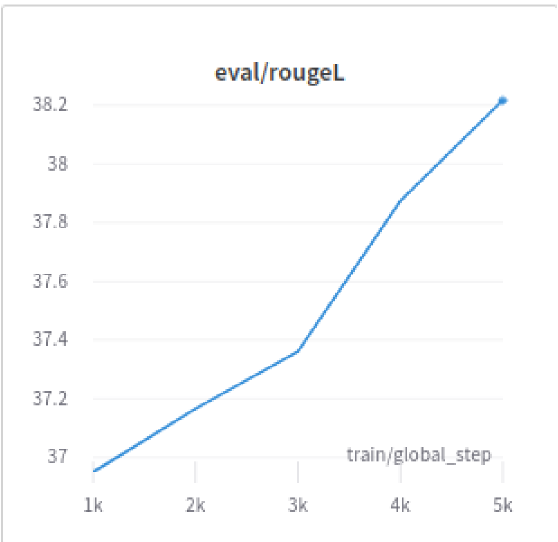
Next, I implemented the T5 model which turned out to be much easier than implementing the previous model. First I created a script to read the dataset and split it into training and testing data then save the dataset object to disk. After this, in the training script, I load the dataset and tokenise it using the 'google/t5-v1_1-base' tokeniser for the T5 model. I then defined a ROUGE evaluation function which I did not write myself, instead it is from the hugging face website and a link to it is available as reference 5 in the references section of this report. Lastly, I load the 'google/t5-v1_1-base' model and define hyper parameters.

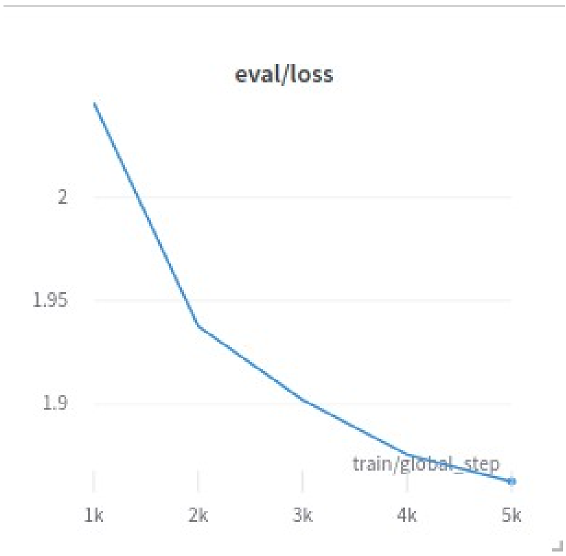
I trained the model with the following parameters:

Batch size	8
Epochs	3
Learning rate	5.6e-5
Weight decay	0.01
Max abstract length	512
Max title length	128









After the model was trained I created a 'Generator' class for generating the titles from abstracts. It works similarly to the Generator class for the previous model in that it takes a abstracts, temperature, number of beams, number of titles to generate and max length arguments then tokenises the abstract and passes it to the model. The model output tokens are then converted back to words and the titles are returned.

Evaluating models

Now that I had two trained models it was time to create a script which calculates BLEU, ROUGE and METEOR scores to evaluate the titles generated by each model. To do this I used the 'nltk' library[10] for evaluating BLEU and METEOR and the 'rouge_score' library[12] to evaluate ROUGE.

```
# Calculate BLEU and METEOR scores
for reference, candidate in zip(ground_truth_tokens, generated_titles_tokens):
    bleu_scores.append(bleu_score.sentence_bleu([reference], candidate, smoothing_function=smoothing))
    meteor_scores.append(meteor_score.single_meteor_score(reference, candidate))

# Calculate ROUGE-L
for reference, candidate in zip(ground_truth, generated_titles):
    rouge_scores.append(rouge_scorer.score(reference, candidate)['rougeL'].fmeasure)
```

After creating the script I ran it on each model and the results can be seen below.

```
Average BLEU score: 0.023550
Average ROUGE-L score: 0.1680
Average METEOR score: 0.0999
```

BiLSTM model scores

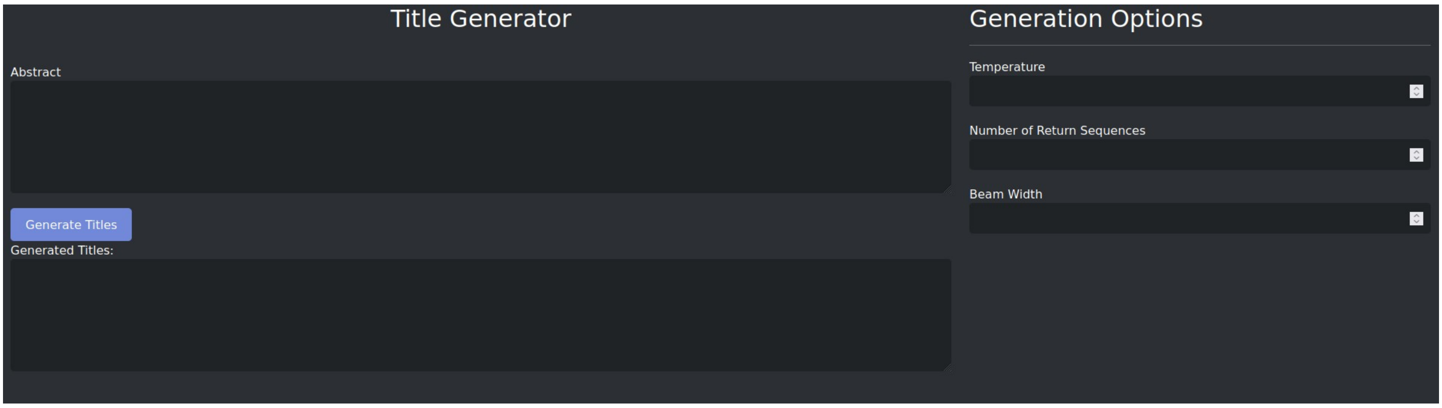
```
Average BLEU score: 0.108750
Average ROUGE-L score: 0.3742
Average METEOR score: 0.3312
```

T5 model scores

The scores clearly show that the T5 model outperforms the BiLSTM by far.

Frontend Interface

With the models now ready, I now used Flask to implement a frontend for generating titles using the T5 model. First I created a basic page using HTML and CSS which has a input text area for abstracts, options for selecting temperature, beam width and number of titles to generate and finally a output text area as can be seen in the screenshot below.


Title Generator
Abstract
Generate Titles
Generated Titles:
Generation Options
Temperature
Number of Return Sequences
Beam Width

I then wrote the flask code necessary to make the system work. The system works by having two flask servers, one which serves the web page, waits for user input and then makes a API call to the second server to generate titles. The second flask server is responsible for loading the model and when it receives a API call it will use the model to generate titles and return them to the frontend server, which in turn displays them to the user.

The reason why I went with this modular architecture instead of just having one flask server is because it has multiple benefits. Firstly, since generating the titles is computationally intensive, it allows for the backend to be hosted on a separate machine specially designed for machine learning where as the frontend can be hosted on a much less powerful server. Secondly, it allows for more models to be added with ease in the future. And lastly, it allows for easy scaling if the system was to be used and had high demand more backend servers could be spun up and the frontend server could distribute its API calls among them.

Results and Evaluation

In this section, I will present the results and evaluate the two deep learning models and compare them to OpenAI's ChatGPT using the GPT-3.5 model. The primary purpose of this section is to analyse the performance of both models and draw conclusions about their suitability for the task of generating titles from abstracts.

First I use the evaluation scripts that I created to calculate the ROUGE-L, BLEU and METEOR scores for the BiLSTM and T5 models. I use a testing dataset of 1000 titles and abstracts and a range of values for temperature and beam width. The results can be seen below.

Temperature	Beam width	BiLSTM ROUGE-L	T5 ROUGE-L	BiLSTM BLEU	T5 BLEU	BiLSTM METEOR	T5 METEOR
1	1	0.0797	0.4043	0.0097	0.1281	0.0346	0.3643
5	1	0.0805	0.4043	0.0107	0.1281	0.0334	0.3643
10	1	0.0298	0.4043	0.0044	0.1281	0.0127	0.3643
32	1	0.0489	0.4043	0.0072	0.1281	0.0208	0.3643
1	5	0.0858	0.4074	0.0117	0.1278	0.0428	0.3620
5	5	0.0838	0.4074	0.0115	0.1278	0.0422	0.3620
10	5	0.0844	0.4074	0.0118	0.1278	0.0427	0.3620
32	5	0.0854	0.4074	0.0116	0.1278	0.0426	0.3620
1	16	0.0207	0.4024	0.0207	0.1234	0.0967	0.3607
5	16	0.1686	0.4024	0.0209	0.1234	0.0970	0.3607
10	16	0.1698	0.4024	0.0203	0.1234	0.0971	0.3607
32	16	0.1644	0.4024	0.0202	0.1234	0.0952	0.3607
1	32	0.1552	0.4038	0.0212	0.1246	0.0923	0.3618
5	32	0.1571	0.4038	0.0210	0.1246	0.0926	0.3618
10	32	0.1573	0.4038	0.0216	0.1246	0.0933	0.3618
32	32	0.1582	0.4038	0.0212	0.1246	0.0928	0.3618

The results shows that the titles generated by the T5 model always outperform the BiLSTM model in the evaluation metrics used. For the BiLSTM model, scores increase with larger beam width whereas the temperature seems to have little effect. For the T5 model, the scores are not affected at all by the temperature and only very slightly affected by the beam width.

From these results it is clear that the T5 models has much better performance than the BiLSTM in all the metrics used. However, these metrics alone are not enough to compare performance so now I will do some manual evaluation and compare the titles generated by each model as well as comparing to titles generated by ChatGPT using the GPT-3.5 model.

For generating titles with ChatGPT I used the following prompt: "I will give you abstracts from scientific papers. I want you to generate a title for each one of them."

1.

Real title: Attention Is All You Need

Real abstract:

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data. [6]

Title generated by BiLSTM: learning decoding for infinitedimensional with stacked networks translation

Validity:

The title is not valid. It seems to generate a vague and somewhat incoherent set of terms related to the abstract, but it doesn't accurately represent the main theme of the abstract.

Correct words or phrases:

While the title includes terms like "learning", "decoding", and "translation" which are relevant to the abstract, the phrase "infinitedimensional with stacked networks" doesn't make sense in this context.

General comment:

The title lacks coherence and doesn't capture the essence of the abstract, which is about the advantages of the Transformer architecture based solely on attention mechanisms.

Title generated by T5: The Transformer: A Simple Network Architecture Based on Attention Mechanisms

Validity:

The title is valid and accurately represents the abstract.

Correct words or phrases:

The title includes the words "Transformer", "Simple Network Architecture", and "Attention Mechanisms", which are key elements in the abstract.

General comment:

This title effectively summarizes the main theme of the abstract - the introduction and benefits of the Transformer architecture that utilizes attention mechanisms. It's concise, coherent, and clearly related to the abstract.

Title generated by ChatGPT: Transformer: A Simple Attention-Based Architecture for Dominant Sequence Transduction Models with Improved Quality and Efficiency

Validity:

The title is valid and accurately represents the abstract.

Correct words or phrases:

All of the words in the title are correctly used and are related to the abstract.

General comment:

This title summarizes the abstract well however the title is quite long.

2.

Real title: SparseGNV: Generating Novel Views of Indoor Scenes with Sparse Input Views

Real abstract:

We study to generate novel views of indoor scenes given sparse input views. The challenge is to achieve both photorealism and view consistency. We present SparseGNV: a learning framework that incorporates 3D structures and image generative models to generate novel views with three modules. The first module builds a neural point cloud as underlying geometry, providing contextual information and guidance for the target novel view. The second module utilizes a transformer-based network to map the scene context and the guidance into a shared latent space and autoregressively decodes the target view in the form of discrete image tokens. The third module reconstructs the tokens into the image of the target view. SparseGNV is trained across a large indoor scene dataset to learn generalizable priors. Once trained, it can efficiently generate novel views of an unseen indoor scene in a feed-forward manner. We evaluate SparseGNV on both real-world and synthetic indoor scenes and demonstrate that it outperforms state-of-the-art methods based on either neural radiance fields or conditional image generation. [13]

Title generated by BiLSTM: generative vision deblurring transformers

Validity:

This title is not valid in relation to the abstract. It does not capture the essence of the paper which revolves around generating novel views of indoor scenes.

Correct words or phrases:

The term "generative" is relevant as the paper deals with a generative task. However, "vision deblurring transformers" does not align with the paper's focus.

General comment:

The title lacks a clear connection to the key points in the abstract. The focus of the abstract is on the novel view generation of indoor scenes using the SparseGNV model, which is not reflected in the title.

Title generated by T5: Searching for novel views of indoor scenes using SparseGNV

Validity:

This title is valid and accurately represents the abstract.

Correct words or phrases:

The terms "novel views", "indoor scenes", and "SparseGNV" are directly related to the abstract.

General comment:

This title concisely summarizes the primary focus of the abstract - the use of the SparseGNV model for generating novel views of indoor scenes. It's coherent and closely aligned with the abstract.

Title generated by ChatGPT: SparseGNV: Generating Photorealistic and Consistent Novel Views of Indoor Scenes using 3D Structures and Image Generative Models

Validity:

The generated title accurately reflects the abstract, capturing the goal of generating novel indoor views with photorealism and consistency.

Correct words or phrases:

"SparseGNV", "generating novel views", "indoor scenes", "3D structures", and "image generative models" are appropriately included from the abstract.

General comment:

The title effectively summarises the abstract's main points but, it misses out on the aspect of comparison to state-of-the-art methods.

3.

Real title: Is dataset condensation a silver bullet for healthcare data sharing?

Real abstract:

Safeguarding personal information is paramount for healthcare data sharing, a challenging issue without any silver bullet thus far. We study the prospect of a recent deep-learning advent, dataset condensation (DC), in sharing healthcare data for AI research, and the results are promising. The condensed data abstracts original records and irreversibly conceals individual-level knowledge to achieve a bona fide de-identification, which permits free sharing. Moreover, the original deep-learning utilities are well preserved in the condensed data with compressed volume and accelerated model convergences. In PhysioNet-2012, a condensed dataset of 20 samples can orient deep models attaining 80.3% test AUC of mortality prediction (versus 85.8% of 5120 original records), an inspiring discovery generalised to MIMIC-III and Coswara datasets. We also interpret the inhere privacy protections of DC through theoretical analysis and empirical evidence. Dataset condensation opens a new gate to sharing healthcare data for AI research with multiple desirable traits. [14]

Title generated by BiLSTM: attentionbased mask sharing hand prediction plants

Validity:

This title is a collection of unrelated terms and does not reflect the content of the abstract.

Correct words or phrases:

The term "sharing" is relevant to the abstract, but the other words (attentionbased, mask, hand prediction, plants) do not align with the key points of the abstract.

General comment:

The title lacks coherance and doesn’t accurately represent the abstract

Title generated by T5: Dataset Condensation in Sharing Healthcare Data for AI Research

Validity:

This title is valid as it accurately captures the main theme of the abstract.

Correct words or phrases:

The terms "Dataset Condensation", "Sharing Healthcare Data", and "AI Research" are directly related to the abstract.

General comment:

The title effectively summarises the main idea of the abstract - the exploration of dataset condensation as a potential solution for healthcare data sharing in AI research. It's concise, coherent, and clearly connected to the abstract.

Title generated by ChatGPT: Dataset Condensation: Promising Prospect for Healthcare Data Sharing in AI Research with Privacy Preservation and Preserved Utility

Validity:

The generated title accurately portrays the content of the abstract, emphasizing dataset condensation's potential for healthcare data sharing in AI research while preserving privacy.

Correct words or phrases:

The title correctly includes phrases such as "dataset condensation", "healthcare data sharing", "AI research", "privacy preservation", and "preserved utility".

General comment:

The title summarise the abstract well. The generated title is longer than the ones generated by the other models.

4.

Real title: Data-inspired modeling of accidents in traffic flow networks

Real abstract:

We consider hyperbolic partial differential equations (PDEs) for a dynamic description of the traffic behavior in road networks. These equations are coupled to a Hawkes process that models traffic accidents taking into account their self-excitation property which means that accidents are more likely in areas in which another accident just occured. We discuss how both model components interact and influence each other. A data analysis reveals the self-excitation property of accidents and determines further parameters. Numerical simulations using risk measures underline and conclude the discussion of traffic accident effects in our model. [15]

Title generated by BiLSTM: stochastic of and shortterm in networks network

Validity:

This title is not valid as it doesn't provide a coherent or meaningful summary of the abstract.

Correct words or phrases:

The term "networks" is relevant, as the abstract discusses traffic behavior in road networks. However, the rest of the terms do not clearly relate to the content of the abstract.

General comment:

The title lacks coherence and doesn't accurately capture the key theme of the abstract, which focuses on the use of hyperbolic partial differential equations and a Hawkes process for modeling traffic behavior and accidents.

Title generated by T5: Hyperbolic partial differential equations for traffic behavior in road networks

Validity:

This title is valid and aligns with the primary topic of the abstract.

Correct words or phrases:

The terms "Hyperbolic partial differential equations", "traffic behavior", and "road networks" are directly related to the abstract content.

General comment:

The title effectively summarizes a key aspect of the abstract - the use of hyperbolic partial differential equations to model traffic behavior in road networks. However, it does not capture the aspect of modeling traffic accidents using a Hawkes process.

Title generated by ChatGPT: Modeling Traffic Behavior and Accident Dynamics in Road Networks: Interactions and Analysis of Hyperbolic PDEs Coupled with Hawkes Processes

Validity:

The title accurately represents the abstract, which focuses on modeling traffic behaviour.

Correct words or phrases:

"hyperbolic PDEs", "Hawkes processes", and "traffic behavior" are present in the given abstract.

General comment:

While accurate, the title is a bit lengthy. A more concise title could convey the same information more effectively.

5.

Real title: Nonlinear Fourier spectrum characterization of time-limited signals

Real abstract:

Addressing the optical communication systems employing the nonlinear Fourier transform (NFT) for the data modulation/demodulation, we provide an explicit proof for the properties of the signals emerging in the so-called b-modulation method, the nonlinear signal modulation technique that provides explicit control over the signal extent. We present details of the procedure and related rigorous mathematical proofs addressing the case where the time-domain profile corresponding to the b-modulated data has a limited duration, and when the bound states corresponding to specifically chosen discrete solitonic eigenvalues and norming constants, are also present. We also prove that the number of solitary modes that we can embed without violating the exact localisation of the time-domain profile, is actually infinite. Our theoretical findings are illustrated with numerical examples, where simple example waveforms are used for the b-coefficient, demonstrating the validity of the developed approach. We also demonstrate the influence of the bound states on the noise tolerance of the b-modulated system. [16]

Title generated by BiLSTM: a rules of the nonlinear and transform with

Validity:

The title is not valid as it doesn't provide a coherent or meaningful summary of the abstract.

Correct words or phrases:

The terms "nonlinear" and "transform" are relevant, as the abstract discusses the nonlinear Fourier transform. However, the rest of the terms do not clearly relate to the content of the abstract.

General comment:

The title lacks coherence and doesn't accurately capture the key theme of the abstract

Title generated by T5: Nonlinear Fourier transform for data modulation/demodulation

Validity:

This title is valid and aligns with the main theme of the abstract.

Correct words or phrases:

The terms "Nonlinear Fourier transform", "data", and "modulation/demodulation" directly relate to the abstract content.

General comment:

The title effectively summarizes a key aspect of the abstract - the use of the nonlinear Fourier transform for data modulation/demodulation.

Title generated by ChatGPT: Analysis and Properties of B-Modulation in Nonlinear Fourier Transform-Based Optical Communication Systems

Validity:

The generated title is valid, accurately representing the focus on b-modulation in NFT-based systems.

Correct words or phrases:

"Nonlinear Fourier Transform", "b-modulation", and "optical communication systems" are correctly used.

General comment:

The title effectively captures the abstract's main points.

6.

Real title: High-Order Large-Eddy Simulations of a Wind Turbine in Ducted and Open-Rotor Configurations

Real abstract:

High-order large-eddy simulations are performed to study the performance and flow field of a ducted wind turbine operating at different tip speed ratios. To evaluate the effects of the duct, simulations with the same tip speed ratios are also performed on the corresponding open-rotor turbine. It is found that the ducted turbine consistently obtains higher power outputs than the open-rotor counterpart, and the duct itself enhances flow turbulence and blade trailing-edge vortices but weakens tip and hub vortices. Flow bifurcation is observed at the largest tip speed ratio and is identified to be caused by blade blockage effects. Comparative simulations are also performed on both turbines under different yaw angles. It is noticed that the ducted configuration is insensitive to small yaw angles and maintains higher power outputs than the open-rotor configuration at all yaw angles. Moreover, it is observed that the wakes of both configurations recover more quickly as the yaw angle increases. [17]

Title generated by BiLSTM: reconstruction of profile the obtained on in shape accuracy from at frequency and principal

Validity:

The title is not valid as it does not make sense or provide a coherent summary of the abstract.

Correct words or phrases:

None of the phrases or words in the title directly relate to the content of the abstract.

General comment:

The title lacks coherence and does not accurately capture the key theme of the abstract, which revolves around the performance and flow field of a ducted wind turbine in comparison to an open-rotor turbine.

Title generated by T5: High-order large-eddy simulations of a ducted wind turbine

Validity:

This title is valid and aligns with the primary topic of the abstract.

Correct words or phrases:

The terms "High-order large-eddy simulations" and "ducted wind turbine" are directly related to the abstract content.

General comment:

The title effectively summarizes a key aspect of the abstract - the use of high-order large-eddy simulations to study the performance of a ducted wind turbine. However, it doesn't capture the comparison aspect with the open-rotor turbine.

Title generated by ChatGPT: Performance and Flow Field Analysis of Ducted Wind Turbines through High-Order Large-Eddy Simulations

Validity:

The generated title accurately reflects the abstract's content, focusing on the performance analysis of ducted wind turbines.

Correct words or phrases:

"High-order large-eddy simulations", "performance", "flow field", and "ducted wind turbines" are correctly used.

General comment:

The title effectively captures the main themes of the abstract. However, it could mention the comparison with open-rotor turbines for better completeness.

7.

Real title: The kinetic theory of mutation rates

Real abstract:

The Luria--Delbrück mutation model is a cornerstone of evolution theory and has been mathematically formulated in a number of ways. In this paper we illustrate how this model of mutation rates can be derived by means of classical statistical mechanics tools, in particular by modeling the phenomenon resorting to methodologies borrowed from classical kinetic theory of rarefied gases. The aim is to construct a linear kinetic model that can reproduce the Luria--Delbrück distribution starting from the elementary interactions that qualitatively and quantitatively describe the variation of mutated cells. The kinetic description is easily adaptable to different situations and makes it possible to clearly identify the differences between the elementary variations leading to the formulations of Luria--Delbrück, Lea--Coulson, and Kendall, respectively. The kinetic approach additionally emphasizes basic principles which not only help to unify existing results but also allow for useful extensions. [18]

Title generated by BiLSTM: the grid

Validity:

This title is not valid and does not reflect the content of the abstract. It's very short and does not provide any context or connection to the subject matter.

Correct words or phrases:

None of the terms in the title relate to the content of the abstract.

General comment:

The title does not have any meaning

Title generated by T5: A linear kinetic model for the Luria--Delbrück mutation model

Validity:

This title is valid and aligns with the main theme of the abstract.

Correct words or phrases:

The terms "linear kinetic model" and "Luria--Delbrück mutation model" are directly related to the abstract content.

General comment:

The title effectively summarizes a key aspect of the abstract - the construction of a linear kinetic model to reproduce the Luria--Delbrück distribution. It's concise, coherent, and clearly connected to the abstract.

Title generated by ChatGPT: A Kinetic Approach to Deriving the Luria-Delbrück Mutation Model: Unifying Principles and Extensions in Evolutionary Theory

Validity:

The generated title accurately represents the content of the abstract, focusing on deriving the Luria-Delbrück mutation model using a kinetic approach.

Correct words or phrases:

"Kinetic approach", "Luria-Delbrück mutation model", and "evolutionary theory" are properly used.

General comment:

The generated title summarises the abstract well. The title is much longer than the real title as well as the other generate titles.

8.

Real title: Algebraic network reconstruction of discrete dynamical systems

Real abstract:

We present a computational algebra solution to reverse engineering the network structure of discrete dynamical systems from data. We use monomial ideals to determine dependencies between variables that encode constraints on the possible wiring diagrams underlying the process generating the discrete-time, continuous-space data. Our work assumes that each variable is either monotone increasing or decreasing. We prove that with enough data, even in the presence of small noise, our method can reconstruct the correct unique wiring diagram. [19]

Title generated by BiLSTM: global completion of for the wavelets with linear

Validity:

This title is not valid as it does not provide a coherent or meaningful summary of the abstract.

Correct words or phrases:

None of the terms in the title directly relate to the content of the abstract.

General comment:

The title is a collection of unrelated words and lacks coherence

Title generated by T5: A computational algebra solution to reverse engineering discrete dynamical systems from data

Validity:

This title is valid and aligns with the main theme of the abstract.

Correct words or phrases:

The terms "computational algebra solution", "reverse engineering", "discrete dynamical systems", and "from data" are directly related to the abstract content.

General comment:

The title effectively summarizes the key aspect of the abstract - the use of a computational algebra solution for reverse engineering the network structure of discrete dynamical systems from data.

Title generated by ChatGPT: Computational Algebra for Reverse Engineering Network Structure in Discrete Dynamical Systems: Monomial Ideals and Constraint-Based Inference

Validity:

The generated title accurately represents the key points mentioned in the abstract, such as the use of computational algebra for reverse engineering in discrete dynamical systems.

Correct words or phrases:

"Computational algebra", "reverse engineering", "network structure", "discrete dynamical systems" are all terms correctly picked up from the abstract.

General comment:

The generated title gives a comprehensive view of the abstract content, effectively summarizing the main method and goal of the paper. The mention of "monomial ideals" and "constraint-based inference" also provides specific details about the employed techniques.

9.

Real title: WWFedCBMIR: World-Wide Federated Content-Based Medical Image Retrieval

Real abstract:

The paper proposes a Federated Content-Based Medical Image Retrieval (FedCBMIR) platform that utilizes Federated Learning (FL) to address the challenges of acquiring a diverse medical data set for training CBMIR models. CBMIR assists pathologists in diagnosing breast cancer more rapidly by identifying similar medical images and relevant patches in prior cases compared to traditional cancer detection methods. However, CBMIR in histopathology necessitates a pool of Whole Slide Images (WSIs) to train to extract an optimal embedding vector that leverages search engine performance, which may not be available in all centers. The strict regulations surrounding data sharing in medical data sets also hinder research and model development, making it difficult to collect a rich data set. The proposed FedCBMIR distributes the model to collaborative centers for training without sharing the data set, resulting in shorter training times than local training. FedCBMIR was evaluated in two experiments with three scenarios on BreaKHis and Camelyon17 (CAM17). The study shows that the FedCBMIR method increases the F1-Score (F1S) of each client to 98%, 96%, 94%, and 97% in the BreaKHis experiment with a generalized model of four magnifications and does so in 6.30 hours less time than total local training. FedCBMIR also achieves 98% accuracy with CAM17 in 2.49 hours less training time than local training, demonstrating that our FedCBMIR is both fast and accurate for both pathologists and engineers. In addition, our FedCBMIR provides similar images with higher magnification for non-developed countries where participate in the worldwide FedCBMIR with developed countries to facilitate mitosis measuring in breast cancer diagnosis. We evaluate this scenario by scattering BreaKHis into four centers with different magnifications. [20]

Title generated by BiLSTM: medical automatic lightweight for utilizing selfsupervised omnidirectional platform data chest

Validity:

This title is not valid as it does not provide a coherent or meaningful summary of the abstract.

Correct words or phrases:

The term "medical" and possibly "platform" are relevant but are not used in a way that relates to the content of the abstract.

General comment:

The title lacks coherence and doesn't accurately capture the key theme of the abstract, which revolves around the use of a federated content-based medical image retrieval platform for training CBMIR models.

Title generated by T5: The Federated Content-Based Medical Image Retrieval Platform

Validity:

This title is valid and aligns with the main theme of the abstract.

Correct words or phrases:

The terms "Federated Content-Based Medical Image Retrieval Platform" are directly related to the abstract content.

General comment:

The title effectively summarizes the key aspect of the abstract - the development and use of a federated content-based medical image retrieval platform.

Title generated by ChatGPT: Federated Content-Based Medical Image Retrieval (FedCBMIR): Utilizing Federated Learning for Efficient and Secure Training of CBMIR Models in Histopathology

Validity:

The generated title is accurate, capturing the key aspects of the study: the application of Federated Learning in CBMIR for histopathology.

Correct words or phrases:

The generated title correctly includes "Federated Content-Based Medical Image Retrieval", "Federated Learning", and "CBMIR models in histopathology".

General comment:

The title describes the abstract accurately.

10.

Real title: Simulating local fields in carbon nanotube reinforced composites for infinite strip with voids

Real abstract:

We consider the steady heat conduction problem within a thermal isotropic and homogeneous infinite strip composite reinforced by uniformly and randomly distributed non-overlapping carbon nanotubes (CNTs) and containing voids. We treat the CNTs as thin perfectly conducting elliptic inclusions and assume the voids to be of circular shape and act as barriers to heat flow. We also impose isothermal conditions on the external boundaries by assuming the lower infinite wall to be a heater under a given temperature, and the upper wall to be a cooler that can be held at a lower fixed temperature. The equations for the temperature distribution are governed by the two-dimensional Laplace equation with mixed Dirichlet-Neumann boundary conditions. The resulting boundary value problem is solved using the boundary integral equation with the generalized Neumann kernel. We illustrate the performance of the proposed method through several numerical examples including the case of the presence a large number of CNTs and voids. [21]

Title generated by BiLSTM: heat conduction of discontinuous patch tubes a in with

Validity:

This title is not valid as it doesn't form a coherent sentence or meaningful summary of the abstract.

Correct words or phrases:

The phrase "heat conduction" is relevant and directly related to the abstract content, but the rest of the title doesn't make sense in the context of the abstract.

General comment:

Although the title has some relevance to the abstract, it lacks coherence and doesn't accurately capture the key theme of the abstract

Title generated by T5: The steady heat conduction problem in isotropic and homogeneous infinite strip composites

Validity:

This title is valid and aligns with the main theme of the abstract.

Correct words or phrases:

The terms "steady heat conduction problem", "isotropic and homogeneous", and "infinite strip composites" are directly related to the abstract content.

General comment:

The title effectively summarizes the key aspect of the abstract

Title generated by ChatGPT: Thermal Analysis of Infinite Strip Composites Reinforced with Carbon Nanotubes and Containing Voids: Boundary Integral Equation Approach

Validity:

The title is valid, summarising the abstract well.

Correct words or phrases:

The title accurately uses "Infinite Strip Composites", "Reinforced with Carbon Nanotubes", "Containing Voids", and "Boundary Integral Equation Approach".

General comment:

The title does a good job representing the abstract's content.

From these results, it is clear that the T5 model consistently outperforms the BiLSTM model. The T5 model consistently generated relevant and coherent titles which accurately capture the key themes of the abstracts provided. The titles generated by T5 were valid and used correct words and phrases directly related to the abstract. Even though the titles were not always perfect, they provided a high-level understanding of what the abstract was about.

On the other hand, the BiLSTM model tended to generate titles that were either not coherent or didn’t represent the content of the abstract. In several cases the BiLSTM generated a title that contained some relevant terms however they lacked proper structure to be meaningful. Most titles were completely irrelevant and didn’t contain any words relating to the abstract.

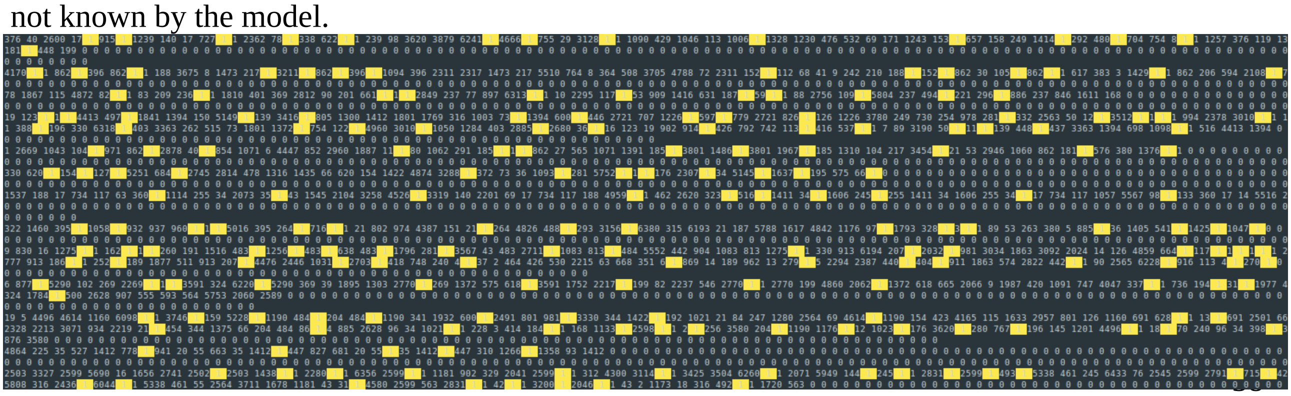
The titles generated by GPT-3.5 were consistently valid and summarised the abstracts well however were also always longer and more detailed than the titles generated by the other models. While a higher level of detail can be beneficial, effective titles should be concise and direct.

In summary, the T5 and GPT-3.5 model had comparable performed, which was much better than the BiLSTM in generating titles for the provided abstracts. This suggests that T5 and GPT-3.5, with their transformer-based architecture, might be better suited for tasks like title generation where context and understanding of the complete input text are important. The T5 however generated shorter and more concise titles than GPT-3.5.

Limitations of implemented BiLSTM model

The model’s performance is determined by a multitude of factors, including its architectural design, the chosen hyperparameters and the quality and size of the dataset. In the following section I will discuss the specific limitations of the current model and illustrate how these may have impacted the model’s ability to generate coherent and accurate titles.

One of the factors which is impacting the performance of this model is the limited vocabulary size used (6500 for the abstracts and 8000 for the titles). This means that the model only understands 6500 unique words in the abstracts and 8000 unique words in titles and any other words are replaced with an unknown token. A larger vocabulary would provide a much richer representation of the text as there would be far fewer unknown tokens. The reason for this constraint is that a larger vocabulary significantly increases the memory usage of the model because the sequences are one hot encoded. This wastes memory because each word is represented by a vector which is the same size as the vocabulary where only one element is 1 and all the others are 0’s. With the vocabulary sized used for this model it was using 64GB of RAM. Here is a sample of the tokenised abstracts with all the unknown words highlighted, this shows many of the words in the abstracts are not known by the model.



The memory usage also meant that the other hyper parameters such as number of layers and hidden size had to be limited. More layers would have allowed the model to learn more complex patterns in the text and possibly improve the accuracy and coherence of the model. However, it would also result in a more complex model which would also add to the memory consumption as well as increasing training time. Similarly, increasing hidden size could have improved the model's performance however also at the cost of increased resources required.

Another factor that limited the performance of the model is that I didn't use a pretrained embedding such as Word2Vec or GloVe. These embeddings provide a dense representation of words based on their semantic meaning, which is pretrained from a large dataset of text data. This would have allowed the model to handle a larger vocabulary as these embeddings provide a compact way to represent words compared to one hot encoding. Using pretrained embeddings also provide the model with more information about the relationships between words, which could significantly improve my model.

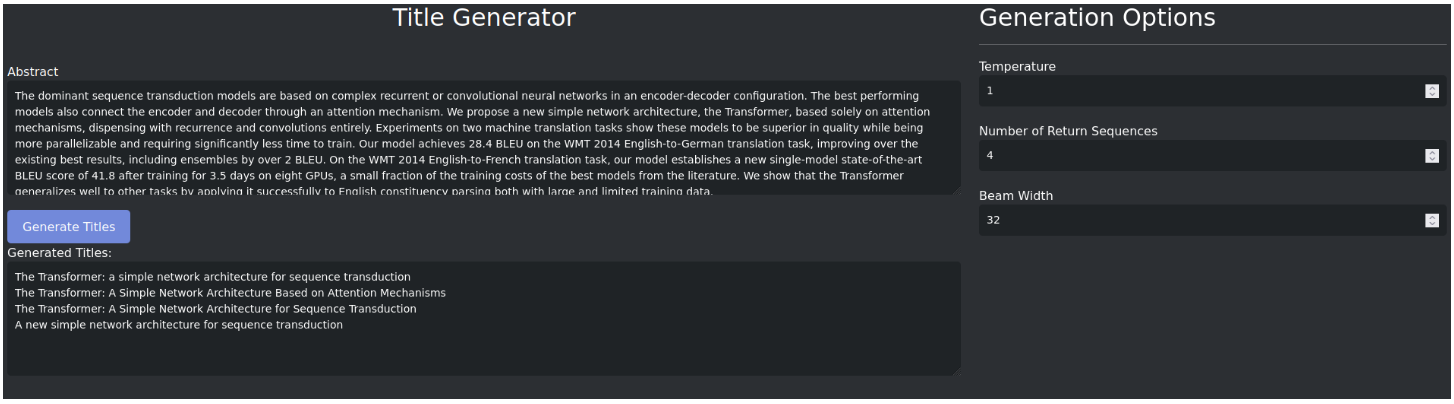
Evaluation of implemented T5 model

The performance of the T5 model was significantly better than the BiLSTM. This is because it is pre-trained on a large dataset of text which gives it a substantial advantage over a model which starts with randomly initiated weights like the BiLSTM model. The pre-training allows the T5 model to start with a good understanding of language which is then fine-tuned to the specific task of title generation. This results in its output being much more coherent sentences and accurately summarise the abstract.

Another of the model's strengths is its transformer architecture which uses a self-attention mechanism to better capture dependencies between words. The self-attention mechanism allows the model to process sequences in parallel instead of sequentially like the BiLSTM. This means that the whole input abstract is processed as a whole rather than one word at a time which allows the model to more accurately capture relationships between words that are far apart.

Evaluation of interface

The interface works well and is efficient and user-friendly. It allows users to easily generate titles with options for temperature, beam width and number of titles. The screenshot below shows the interface being used.



After startup, it takes approximately 12 seconds from pressing the generate button to when a title is returned due to the model needing to be loaded into memory. Consequent titles take approximately 5 seconds. These times were measured with the frontend and backend server running locally and using a GTX 1080 graphics card.

Conclusion

This project aimed to create a system capable of generating titles from paper abstracts using deep learning technique. The overall goal was achieved with the use of the T5 model which demonstrated an excellent capability to generate accurate and cohesive titles. However, some aspects of the project presented challenges and room for future improvement.

I will now go over the objectives set out in my initial plan and evaluate if they have been achieved. The first objective was to ‘Acquire and prepare a large dataset of abstracts and their corresponding titles for model training.’ I achieved this by using the arXiv API and was able to collect a dataset of over two hundred thousand titles and abstracts. However these were from a limited number of subject areas which was a limiting factor in the project.

The next objectives were to implement at least two different deep learning models and then compare their performance. I achieved this by implementing a bidirectional LSTM model and a transformer-based model (T5). I then compared the performance of the two which showed that the T5 was significantly better at generating accurate titles, demonstrating the effectiveness of transformer models in natural language processing.

The last objective was to ‘refine and improve the performance of the most effective technique, if necessary.’ The most effective technique was the T5 model which demonstrated an excellent ability to generate accurate titles after I trained it for the first time. Therefore, further refinement was not necessary.

A significant challenge of this project was with implementing and training the BiLSTM model. The model was very memory inefficient due to the use of one hot encoding and this limited the hyperparameters that I could use to train it as setting them too large would require more memory than my computer has. This caused an issue as the vocabulary size had to be quite small which resulted in many words being unknown by the model and this in turn led to the model not being able to learn from the data very well. I believe this to be the main reason why the model achieved poor results.

In conclusion, the project achieved its aim of developing a deep learning-based system for generating paper titles. The project and the challenges that came with it provided me with valuable insights and knowledge of machine learning and directions for future work.

Future Work

There are several avenues to follow to further enhance this project. Firstly, if I was to continue with this project, I would focus on improving the BiLSTM model as the current implementation has shown underwhelming performance. The main aspect that I would change is to use a pretrained embedding to help the model understand the semantic meaning of words. I would also expand the dataset to include papers from a broader range of subject areas.

From an infrastructure perspective, if I had a large budget for the project, I would build a dedicated machine learning server with multiple GPUs and an abundance of RAM. This would enable me to create more complex models, use larger datasets and accelerate training times.

Another interesting prospect for future work would be to construct a transformer model from scratch. While the T5 transformer model showed impressive performance, developing a custom transformer model would offer me a chance to gain a more profound understanding of the architecture. It could also allow for more customisability and optimisations for the task of generating titles.

I would also investigate using more comprehensive evaluation metrics. While BLEU, ROUGE-L and METEOR served as effective metric, a more comprehensive evaluation approach could be used.

Reflection on Learning

Starting this project with no prior machine learning knowledge, my learning curve was steep but rewarding. Throughout the project, I gained understanding of a broad range of machine learning principles and methodologies, such as text preprocessing, model selection and implementation, and performance evaluation. This project allowed me to explore different architectures like BiLSTM and the transformer based T5 model, each with its unique challenges and learning opportunities.

In the beginning, I learned about the steps involved in gathering and preparing text data for machine learning. I was introduced to text cleaning, stopword removal and tokenisation. The large amount of text data proved to be a challenge to work with as it made the training process very resource and time intensive, often my GPU would be crunching numbers for days on end to train a model.

The largest challenge for me with this project was with the implementation of the BiLSTM model. It took multiple attempts and much troubleshooting to fix errors to get a working model. In stark contrast, implementation of the T5 model was a much smoother process, demonstrating the ease of working with a pre-trained model.

Despite the challenges I faced during my project, I found it very rewarding when I finally figured something out or fixed an error after hours of troubleshooting. Overcoming these challenges not only provided me with a highly sought-after skills in machine learning but also a personal interest which I plan to further develop by taking on other machine learning project in the future.

The high level of independence with this project was a departure from the structured learning environment I had been accustomed to throughout school and university. It fostered a sense of responsibility and self-reliance, skills that are highly valuable for the future.

Looking back at my time at Cardiff university, this project was a fitting conclusion. It encapsulated the essence of my learning journey – challenging, rewarding and filled with growth opportunities. As I move forward, I carry with me not just the technical machine learning skills but also a newfound confidence to navigate and overcome challenges independently.

References

[1] Karen White. 2019. Publications Output: U.S. Trends and International Comparisons. Available at: https://ncses.nsf.gov/pubs/nsb20206/executive-summary [Accessed: 26 April 2023]

[2] Sara Brown. 2021. Machine learning, explained. Available at: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained [Accessed: 26 April 2023]

[3] arXiv. 2023. arXiv API Access. Available at: https://info.arxiv.org/help/api/index.html [Accessed: 28 April 2023]

[4] Bala Priya. 2022. Softmax Activation Function: Everything You Need to Know. Available at: https://www.pinecone.io/learn/softmax-activation/ [Accessed: 28 April 2023]

[5] Hugging Face. 2022. Metrics for text summarization. Available at: https://huggingface.co/learn/nlp-course/chapter7/5#metrics-for-text-summarization [Accessed: 4 April 2023]

[6] Ashish Vaswani et al. 2017. Attention Is All You Need. Available at: https://arxiv.org/abs/1706.03762 [Accessed: 30 April 2023]

[7] Jason Brownlee. 2021. A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Available at: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/ [Accessed: 30 April 2023]

[8] Google Brain Team. 2015. Tensorflow. Available at: https://www.tensorflow.org/ [Accessed: 29 April 2023]

[9] NumPy. 2023. NumPy. Available at: https://numpy.org/ [Accessed: 29 April 2023]

[10] Team NLTK. 2023. Available at: https://www.nltk.org/ [Accessed: 29 April 2023]

[11] Hugging Face. 2023. Transformers. Available at: https://huggingface.co/docs/transformers [Accessed: 29 April 2023]

[12] Google LLC. 2022. rouge-score 0.1.2. Available at: https://pypi.org/project/rouge-score/ [Accessed: 2 May 2023]

[13] Weihao Cheng et al. 2023. SparseGNV: Generating Novel Views of Indoor Scenes with Sparse Input Views. Available at: https://arxiv.org/abs/2305.07024 [Accessed: 11 May 2023]

[14] Yujiang Wang et al. 2023. Is dataset condensation a silver bullet for healthcare data sharing? Available at: https://arxiv.org/abs/2305.03711 [Accessed: 11 May 2023]

[15] Simone Göttlich, Thomas Schillinger. 2023. Data-inspired modeling of accidents in traffic flow networks. Available at: https://arxiv.org/abs/2305.03469 [Accessed: 11 May 2023]

[16] Dmitry Shepelsky et al. 2020. Nonlinear Fourier spectrum characterization of time-limited signals. Available at: https://arxiv.org/abs/1907.01331 [Accessed: 11 May 2023]

[17] Chi Ding et al. 2022. High-Order Large-Eddy Simulations of a Wind Turbine in Ducted and Open-Rotor Configurations. Available at: https://arxiv.org/abs/2206.00313v1 [Accessed: 11 May 2023]

[18] Lorenzo Pareschi, Giuseppe Toscani. 2022. The kinetic theory of mutation rates Available at: https://arxiv.org/abs/2212.00146 [Accessed: 11 May 2023]

[19] Heather A. Harrington, Mike Stillman, Alan Veliz-Cuba. 2022. Algebraic network reconstruction of discrete dynamical systems. Available at: https://arxiv.org/abs/2212.02601 [Accessed: 11 May 2023]

[20] Zahra Tabatabaei et al. 2023. WWFedCBMIR: World-Wide Federated Content-Based Medical Image Retrieval. Available at: https://arxiv.org/abs/2305.03383 [Accessed: 11 May 2023]

[21] Mohamed Nasser et al. 2021. Simulating local fields in carbon nanotube reinforced composites for infinite strip with voids. Available at: https://arxiv.org/abs/2201.00003 [Accessed: 11 May 2023]