Automatic Title Generation for Text with Pre-trained Transformer Language Model

Prakhar Mishra
IIIT Bangalore
India
Email: prakhar.mishra@iiitb.org

Chaitali Diwan
IIIT Bangalore
India
Email: chaitali.diwan@iiitb.org

Srinath Srinivasa
IIIT Bangalore
India
Email: sri@iiitb.ac.in

G.Srinivasaraghavan
IIIT Bangalore
India
Email: gsr@iiitb.ac.in

2021 IEEE 15th International Conference on Semantic Computing (ICSC) | 978-1-7281-8899-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSC50631.2021.00009

***Abstract*—In this paper, we propose a novel approach to Automatic Title Generation for a given text using a pre-trained Transformer Language Model GPT-2. The model proposes an unique approach of generating a pool of candidate titles and selecting an appropriate title among them which is then refined or de-noised to get the final title. The approach consists of a pipeline of three modules namely Generation, Selection and Refinement followed by a Scoring function. The Generation and Refinement modules are based on GPT-2, while the Selection module has a heuristic based approach. The model is able to generate accurate titles in spite of having a smaller corpus of relevant training data due to the fact that the natural language generation capabilities come from the pre-training while the model has to primarily learn task and corpus specific nuances. Additionally, Selection and Refinement modules ensure that the titles are representative of the given text and are semantically and syntactically accurate. We train our model for research paper abstracts from arXiv and evaluate it on three different test sets. Our pipeline shows promising results when evaluated on ROUGE and BLEU metrics against the test sets. In addition, we also perform human evaluation for validating the results generated by our proposed approach.**

Note: The title of this paper was generated automatically by our proposed algorithm from the abstract.

I. Introduction

A representative and an interesting title is invariably the most important aspect of any document. The title is the first, and sometimes the only part of an article or the document that the potential readers will see. While the title has to be catchy to grab the readers attention and entice them to read the full article, it should also accurately portray the content of the document so that readers are not misled by a catchy but not-quite-accurate title. Representative titles also serve as an important feature for information retrieval systems which give more weightage to the keywords that occur in the title.

Title generation is hence an important and challenging problem in the field of NLP (Natural Language Processing). Title generation is a special case of summarisation. Summarisation is the task of shortening the given document into a very short text while preserving the main essence of the underlying topic. Whereas, title generation captures the essence of the document in a couple of words or at the most a sentence while adhering to certain latent features such as linguistic construction.

Automatic summarisation has been studied extensively for decades. Automatic title generation forms a small fraction of these studies. While the previous title generation systems have been able to produce fairly good and relevant titles in some contexts, the problem is still a challenging one when large training data is not available and in niche domains such as scientific and technical articles that have different structural patterns and specific technical vocabulary.

To address this, we propose a novel approach of a supervised generative model for title generation built using a pre-trained GPT-2 (Generative Pre-trained Transformer 2) language model [1]. GPT-2 is a large transformer-based language model trained on 40GB of internet text, with a simple objective of predicting the next word given all the previous words within that sequence. GPT-2 generates synthetic text samples that appear realistic and are coherent continuations of the provided input.

However, generative models such as GPT-2 are probabilistic in nature. Hence when they are used for any specific task, the text generated could be different each time the model is inferenced. In our case, the model could generate different titles each time the model is inferenced. While this has an advantage of producing not one but many semantically accurate and interesting titles, it becomes imperative to choose the best among these titles.

For this, we propose a rule based selection method that chooses the best suitable title among the generated titles. This is followed by another GPT-2 based model that refines or de-noises the chosen title. As a final step, we generate "relevancy scores" for both– selected title and refined title and the title with the highest relevancy score is chosen as the final output title. In case of applications with possible user intervention, we present both selected and refined titles to the end-user with suggested relevancy scores.

We use the dataset of research paper abstracts from arXiv to train our model and then evaluate our model on three different test sets. Experiments using our methodology show promising results when evaluated using ROUGE and BLEU metrics against the hold-out set and two unseen datasets of the same domain. This methodology is easily transferable to any domain of interest with enough relevant data. In addition to the automated evaluation, we also perform human evaluation for validating the results generated by our proposed approach.

The rest of the paper is organised as follows: Section II discusses related work in this area. Section III describes our proposed approach of title generation. Section IV describes the

978-1-7281-8899-7/21/$31.00 ©2021 IEEE
DOI 10.1109/ICSC50631.2021.00009

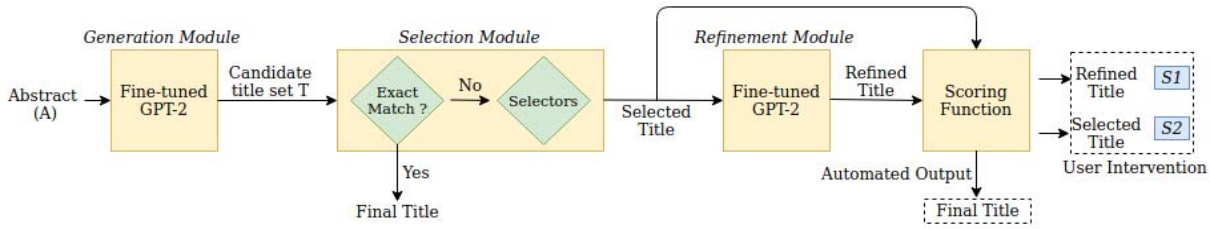

Fig. 1. Pipeline of our Title Generation approach

experiments conducted, results obtained and our observations. Finally, Section V summarises our work and discusses possible directions for future study.

II. Related Work

Automatic generation of titles or headlines for a given text is an interesting and challenging problem pursued by different researchers over many years. We explore and report some of the pioneering works in this area.

A mixed approach of extractive and abstractive methods for shortening text are studied in [2]–[4]. Cédric et al. in [3] discuss an approach where a few important candidate sentences for titling are chosen. From these candidate sentences, important noun phrases are extracted and ranked to create titles. Jin and Hauptmann in [2] describe an approach of first identifying the key contents of the documents in the form of constituents such as words or phrases and then organising them into a grammatical sentence using statistical language models. Tan et al. in [4] propose a coarse-to-fine approach, which first identifies the important sentences of a document and then exploits a multi-sentence summarisation model with hierarchical attention to leverage the important sentences for headline generation.

Colmenares et al. in [5] propose a sequence-prediction technique which models headline generation as a discrete optimization task in a feature-rich space. Putra and Khodra in [6] propose a title generation method that considers the author's intentions. It uses two title generation approaches of template-based and an adaptive K-nearest neighbour.

Approaches in [7]–[10] use encoder-decoder or Transformer architectures. Gu et al. [7] propose a model to generate headline of a cluster of news articles. They use encoder-decoder architecture [11] to extract titles for a single article, followed by a self-voting-based article attention layer to extract salient information shared by multiple articles. Murao et al. in [8] propose a system for generating title with editing support by training an encoder-decoder model from lead and headlines to generate short titles. Lopyrev in [9] uses encoder-decoder recurrent neural network architecture with LSTM unit cells and attention [11] to generate headlines from input news articles. Gavrilov et al. in [10] use Universal Transformer [12] architecture with byte-pair encoding technique that uses self-attention to generate headlines from input news articles.

The above mentioned encoder-decoder or Transformer architectures produce fairly relevant and semantically accurate titles but require a large amount of training data. Whereas our model can be trained using a smaller dataset consisting of few thousand data points.

Another major contribution of our approach is that it produces a pool of candidate titles and provides a method to select the best among them. This is further refined to make syntactic and semantically accurate, and representative titles. An option to choose either selected title or refined title is also presented to the user along with the relevancy score when the model is used in applications that allow user intervention. Apart from the methodology, our experiment also focuses on scientific and technical domains that have challenging patterns and large number of unique concepts, unlike the news datasets used in the above approaches where structural patterns such as leading sentence baselines have a strong say in the titling process.

Few approaches as discussed in [13], [14] study the generation of abstractive summaries using pre-trained GPT-2 (Generative Pre-trained Transformer 2) model similar to our approach, however they generate summaries for the document while we generate titles for a given text which is a specific subset of summarisation and has different challenges.

III. Approach

Our approach to generating the titles for a short document is based on fine-tuning Open AI's pre-trained language model GPT-2 [1]. Fine-tuning the pre-trained models are seen to tremendously improve the performance of any down stream tasks. Hence we choose to use this method in our approach.

Our model of title generation consists of three modules: Generation, Selection and Refinement followed by a Scoring function. Figure 1 shows the pipeline of our title generation approach. The abstract is fed to the Generation module, which generates a pool of titles. From this pool, we sample a candidate set of titles 'T' using Top-k Top-p sampling strategy which are passed to the Selection module. In Selection module, a heuristic based approach is used to select the best title among the sampled titles from the set 'T'. Finally, the chosen title is passed through the Refinement module which de-noises the title by producing a refined title which is syntactically and semantically better.

After generating the refined title, the Scoring function then scores the titles from Selection module and Refinement module based on a Jaccard similarity. This makes our model robust to selecting the final title by attending to the best of both

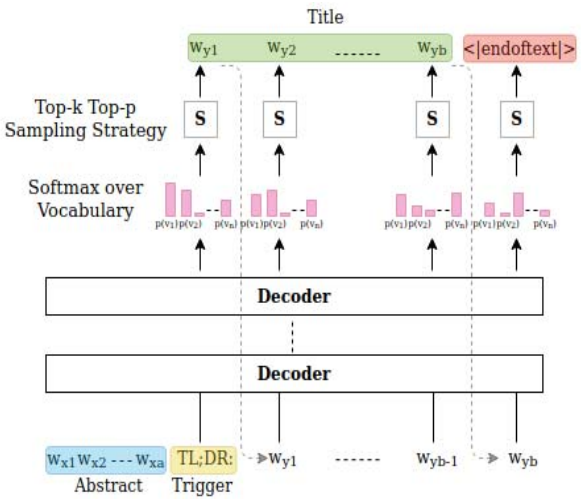

Fig. 2. Illustration of Generation module at Inference time

worlds– between a title comprising newly generated terms, versus a title that contains elements of actual content in the abstract. As mentioned earlier, the model can present both the options to the user along with the relevancy score and the user can choose the title according to their requirement by giving preferred weightage to each of these factors. In an automated pipeline, the title with the higher relevancy score is chosen as the final title.

A. Generation module

We use a medium sized GPT-2 model (345M parameters) as our choice of pre-trained language model. Language models such as GPT-2 can be used in the text-to-text framework simply by concatenating the inputs and the targets by adding a separator word between them. We use the separator $TL;DR:$ that was used for summarisation in the GPT-2 paper by Radford et al. [1], since titling is a sub task of summarisation and we believe that the pre-trained GPT-2 has already learnt the summarisation behaviour for this keyword.

Formally, consider each data-point $d_i(x_i, y_i)$ of the dataset $D(X, Y)$ with 'n' data-points where $x_i \in X$ represents the abstract and $y_i \in Y$ represents the title. Let $x_i = w_{x_1}, w_{x_2}, \ldots, w_{x_a}$ and $y_i = w_{y_1}, w_{y_2}, \ldots, w_{y_b}$ where a and b are the lengths of x_i and y_i respectively. We re-define each data-point by concatenating them with the separator keyword $t = TL;DR:$ and end token $e =< |endoftext| >$ such that data-point is now $d_i = x_i, t, y_i, e$.

For simplicity in the problem formulation, we replace each word/symbol in the data-point with u_i. The data-point thus becomes $d_i = u_1, u_2, ..., u_L$ where length of the data-point $L = a + b + 2$ and $L \leq 1024$.

We maximise the objective as defined in eq. 1

$$P(D) = \prod_{i=1}^{n} \prod_{j=1}^{L} P(u_j | u_1, u_2, .., u_{j-1}; \theta) \tag{1}$$

Here, $P(u_j | u_1 ... u_{j-1}; \theta)$ is the conditional probability of obtaining u_j given the previous tokens $u_1 ... u_{j-1}$, θ represents the model parameters.

We fine-tune GPT-2 language model by optimizing on the cross-entropy loss as defined in eq. 2.

$$Loss = -\sum_{i} p_i \log_2 p'_i \tag{2}$$

where, p'_i is the probability generated by our model for word i and p_i is the ground truth probability of word i.

For decoding purposes, we considered different decoding or sampling techniques such as Greedy, Beam Search [15], Top-k [16] and Top-p (Nucleus) sampling [17]. We know that using Greedy method to sample words results in a deterministic decoding which is undesirable considering our application. Holtzman et al. in [17] discuss disadvantages of using Beam Search and Top-k sampling techniques. The drawback of Beam search is that it is not diversified enough and generates repeated patterns. The disadvantage of using Top-k sampling is that it returns high number of low probability predictions if the distribution is peaked. We see that Top-p sampling technique avoids text degeneration by truncating the unreliable tail of the probability distribution, however it creates a large sample space when the distribution is flat.

Considering the pros and cons of both the cases, we implement a mix version of both the techniques called Top-k Top-p sampling. We first use Top-k sampling where we sample k high softmax probability terms. Then on this set, we use Top-p sampling to get the set of keywords whose cumulative probability mass is at-least p. (The value of k and p are empirically decided and is discussed in Section IV-B.) On this set, we sample the word based on its conditional probability distribution. The Top-k Top-p sampling thus ensures that the sample is bounded with max k terms for flat distributions and minimum cumulative probability mass equal to p for peaked distribution.

During inference phase, we generate a set of candidate titles $T =< t_1, t_2 ... t_n >$ where each title t_i represents the title generated from Generation module. Figure 2 illustrates the Generation module at inference time. As seen in figure, we start generating title words $w_{y_1}, w_{y_2} ...$ after adding the keyword $t = TL;DR:$ at the end of the abstract $w_{x_1}, w_{x_2} ...$ and stop when the end token $< |endoftext| >$ is produced by our model. At every generation timestep, we sample a word using Top-k, Top-p sampling strategy on the softmax probability distribution over the vocabulary. We also implement a post-processing step to address the adjacent word duplication problem to make it syntactically sound.

B. Selection Module

Top-k Top-p sampling technique is expected to produce variety of titles when sampled due to randomness inherited at every time-step. The title generated from GPT-2 model at every inference would be different. This means that some titles might be better than others in terms of linguistic constructs and semantics for representing the input document. Hence,

Selection Criteria	R-1	R-2	R-L
Random	0.373	0.180	0.329
Exact Match + Random	0.375	0.186	0.332
Exact Match + Noun Phrase + Random	0.380	0.187	0.341
Exact Match + Noun Phrase + Semantic Sim	**0.388**	**0.190**	**0.342**

TABLE I
VALIDATION SCORES TO CHOOSE THE COMPONENTS OF THE SELECTION MODULE

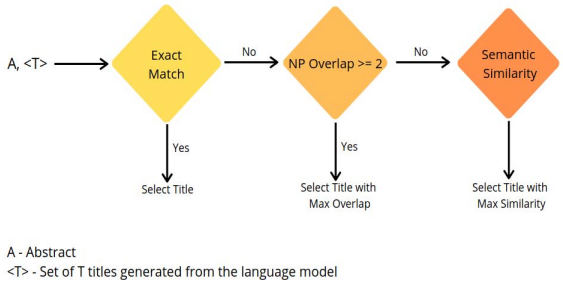

Fig. 3. Selection flow showing the selection of the title from the set of candidate titles

we inference the GPT-2 model 'h' times for the same input abstract to obtain the titles that form the candidate list T (as mentioned in previous sub-section). We experiment with different values of h, and based on the standard deviation of ROUGE scores, we choose the value $h = 5$ for the corpus.

To select the most appropriate title for the given input from the pool of candidate titles, we introduce the Selection Module in the pipeline. We implement three components in Selection module namely Exact Match, Noun Phrase Overlap and Semantic Similarity. To select the appropriate components for this module, we use forward selection procedure which is a popular technique for feature selection in machine learning.

We start off with randomly selecting the component. Then we keep adding each of the three components to the Selection Module one at a time, based on their performance. Table I shows the observed ROUGE-1,2,L scores on validation set while choosing components during forward selection technique. Reported numbers are the average scores over 5 runs. The chosen components then form the 3-step greedy selection pipeline as described in Figure 3.

The first component of the Selection module called the Exact Match checks if the generated title matches exactly with the sentence present in the abstract. Our rationale to check for exact occurrence is that such a title would be grammatically correct, fluent and would also portray the central theme of the input document. However, the exact matching of title with abstract forms a very small percentage of the titles generated, especially in the scientific domain like ours. In case of our corpus of arXiv, only 4% of the abstracts of the training dataset had a title whose text was already present in the abstract. Nevertheless our approach first checks for it due to the above mentioned advantages.

The second component of the Selection module is based on the Noun Phrase (NP) overlap between title and abstract which pushes our selection procedure towards extractive nature. This rule helps us in selecting title that is syntactically closer to the input document rather than selecting a title that is totally diverse. We apply Grid search to tune the NP overlap threshold parameter.

In the last component of the Selection module, we vectorise the title and the input document using pre-trained Sentence Transformers [18] and then measure the semantic closeness between them using cosine similarity. Title with the highest similarity is selected.

Algorithm. 1 outlines the pseudo code for the components in the pipeline of the rule based Selection module as discussed above.

Algorithm 1 Selection Module Pseudo code

$syn, sem \leftarrow emptydictionary$
for $i = 1, 2, \ldots T$ **do**
  if $title_i$ in abstract **then**
    return $title_i$
  end if
  $syn[title_i] \leftarrow NP(title_i, abstract)$
  $v_t, v_a = Vectorise(title_i, abstract)$
  $sem[title_i] \leftarrow Cosine(v_t, v_a)$
end for
if $max(syn) \geq 2$ **then**
  $title_k = max(syn)$
  return $title_k$
end if
$title_k = max(sem)$
return $title_k$

C. Refinement Module

In the Refinement module, we fine-tune medium sized GPT-2 language model that is trained separately and the sequence is formatted similar to the Generation module, with the separator word being $< to - denoise >$.

For fine-tuning this model, we created the dataset by considering the sampling titles from the output of the Generation module as the inputs to the Refinement module, and the original titles as the expected output. This model is supposed to learn the mapping from the input title to the original title and push the generation capabilities towards the ground truth. Training of this component involved optimizing on the similar objective as the Generation module. However, unlike in the Generation module, here we used the Greedy decoding strategy to sample the words at each time-step. This is because we desire a deterministic output from the Refinement module.

D. Scoring Function

The Scoring function is the last step in the pipeline that calculates the relevancy scores for both the titles from the Selection module and Refinement module. This is calculated using Jaccard Similarity over Noun Phrase set between titles and the input abstract. A smoothing value of 0.0001 is also

Abstract: Learning the distance metric between pairs of examples is of great importance for visual recognition, especially for person re-identification (Re-Id). Recently, the contrastive and triplet loss are proposed to enhance the discriminative power of the deeply learned features, and have achieved remarkable success. As can be seen, either the contrastive or triplet loss is just one special case of the Euclidean distance relationships among these training samples. Therefore, we propose a structured graph Laplacian embedding algorithm, which can formulate all these structured distance relationships into the graph Laplacian form. The proposed method can take full advantages of the structured distance relationships among these training samples, with the constructed complete graph. Besides, this formulation makes our method easy-to-implement and super-effective. When embedding the proposed algorithm with the softmax loss for the CNN training, our method can obtain much more robust and discriminative deep features with inter-personal dispersion and intra-personal compactness, which is essential to person Re-Id. We illustrate the effectiveness of our proposed method on top of three popular networks, namely AlexNet, DGDNet and ResNet50, on recent four widely used Re-Id benchmark datasets. Our proposed method achieves state-of-the-art performances.
Actual: Deep Feature Learning via Structured Graph Laplacian Embedding for Person Re-Identification
Title-1: A Structured Graph Laplacian Embedding for Deep Person Re-Id / **RS**: 0.03
Title-2: Structured Graph Laplacian Embedding of Contrastive and Triplet Losses
Title-3: A Structured Graph Algorithm for Deep Person Re-Id
Title-4: Structured Distance Relations among Training Examples
Title-5: Structured Graph Laplacian Embedding for Deep Person Re-Id
Refined Title: Structured Graph Laplacian Embedding for Person Re-identification / **RS**: 0.07

TABLE II
OUTPUT FROM SELECTION AND REFINEMENT MODULE. HERE RS IS THE RELEVANCY SCORE

added to avoid the cases of zero Jaccard similarity. Refer equation 3.

$$relevancy\ score(t) = \frac{NP(t) \cap NP(A) + 0.0001}{NP(t) \cup NP(A)} \quad (3)$$

where, t, NP and A refers to title, Noun Phrase and Abstract respectively.

The reason we use the Noun Phrase Overlap metric again in the final step to calculate the relevancy score is because the Refinement module could add some relevant noun phrases or could remove the redundant noun phrases from the title. We can see this through a random example showcased in Table II where the outputs for every module are shown. The table shows abstract, actual title, five titles from Generation module, selected title from Selection output (shown in green color) and refined title for the selected title processed by the Refinement module. The Noun Phrase Overlaps are marked in red color. We can observe that the noun phrase "Person Re-Identification" was added to the title after Refinement step. It replaced the word "Re-Id" which does not make semantic sense, with the correct noun phrase "Re-Identification". The Refinement module was able to correctly identify the replacement word. The relevancy score (RS) thus increased for this title leading to its selection in case of the automated pipeline. This re-iterates the necessity of Noun Phrase Overlap in the last phase of final title selection.

IV. EXPERIMENTS

In this section, we describe the dataset, the details of the implementation of our model and baseline algorithms.

A. Dataset

We train our model using **arXiv dataset**[1]. The arXiv dataset consists of json objects of 41k research papers from domains such as ML, CL, NER, AI and CV from 1980 to 2018. We consider only the abstract and the titles from the dataset which form the input and output pairs for our generative model. We split our data into training set (75%), testing set (25%) and validation set (5% samples from training set).

Datasets	#Docs	Avg. Abs Len	Avg. Title Len	Compress %
arXiv	41k	150.58	8.71	6.71
ACL	10.8k	116.08	9.38	9.22
ICMLA	448	155.26	10.04	7.06

TABLE III
DATASET STATISTICS

We also validate our model on other unseen datasets like **ACL data** (Association for Computational Linguistics) [19] and **ICMLA data** (International Conference on Machine Learning and Applications) [20].

Table III lists some important statistical properties for all the three datasets. The lengths are at word level. Compress % indicates the ratio of title length to summary length.

B. Implementation Details

To fine-tune the medium sized GPT-2 model as part of our Generation module and Refinement module, we used HuggingFace Transformers[2] library in PyTorch[3]. All the necessary hyper-parameters mentioned in Table IV were selected based on the best performing parameters on the validation set. Also, we experimented with various values of k and p for our Top-k Top-p sampling and found k=2 and p=0.8 to give best scores on validation split. The model was trained on a free instance of Google Colab[4] and it took roughly 10 hrs on a single server.

Table VI refers to progressive experimental scores over couple of experimental models that we tried before settling for the final model. All the scores marked with $*$ are average scores over 5 runs on the test split.

C. Baselines

We compare our results with multiple algorithms as follows:
PREFIX In this method, we choose the first sentence in the

[1]arXiv dataset is available at https://tinyurl.com/y9pu6xyp

[2]HuggingFace Transformers - https://github.com/huggingface/transformers
[3]PyTorch - https://pytorch.org/
[4]Google Colab - https://colab.research.google.com/

Parameter	Value
Batch Size	32
Epochs	4
Loss	Cross Entropy
Learning Rate	2e-05
LR Schedule	Cosine with Restart

TABLE IV
MODEL PARAMETERS

Model	Humans		
	Selection	Refinement	Either/Neither
Selection	13	2	0
Refinement	4	6	0

TABLE V
CONFUSION MATRIX FOR HUMAN EVALUATION

	arXiv						
Our Method	**ROUGE**			**BLEU**			
	1	2	L	1	2	3	4
∗ Generation	0.336	0.154	0.306	0.324	0.219	0.179	0.102
∗ Generation + PP	0.368	0.175	0.321	0.348	0.238	0.188	0.109
∗ Generation + PP + Selection	**0.376**	**0.188**	**0.336**	**0.358**	**0.250**	**0.209**	**0.124**
	ACL						
∗ Generation	0.296	0.113	0.268	0.280	0.173	0.136	0.072
∗ Generation + PP	0.328	0.134	0.295	0.317	0.201	0.162	0.090
∗ Generation + PP + Selection	**0.340**	**0.156**	**0.308**	**0.321**	**0.219**	**0.187**	**0.108**
	ICMLA						
∗ Generation	0.361	0.159	0.319	0.315	0.209	0.169	0.101
∗ Generation + PP	0.384	0.181	0.346	0.334	0.227	0.176	0.097
∗ Generation + PP + Selection	**0.404**	**0.189**	**0.356**	**0.395**	**0.271**	**0.216**	**0.123**

TABLE VI
ROUGE AND BLEU SCORES FOR ALL THREE DATASETS BASED ON INCREMENTAL EXPERIMENTS OF OUR METHOD. HERE, PP STANDS FOR POST-PROCESSING

	arXiv						
Method	**ROUGE**			**BLEU**			
	1	2	L	1	2	3	4
PREFIX	0.228	0.099	0.206	0.160	0.103	0.091	0.047
TextRank	0.137	0.039	0.130	0.083	0.043	0.036	0.015
LexRank	0.204	0.08	0.186	0.131	0.079	0.067	0.033
∗ Bi-GRU w/ Attention	0.211	0.045	0.162	0.250	0.189	0.163	0.132
∗ Zero-Shot GPT-2	0.123	0.044	0.170	0.058	0.033	0.030	0.013
∗ Our Method	**0.376**	**0.188**	**0.336**	**0.358**	**0.250**	**0.209**	**0.124**
	ACL						
PREFIX	0.234	0.099	0.213	0.154	0.097	0.086	0.044
TextRank	0.142	0.038	0.133	0.087	0.044	0.037	0.015
LexRank	0.221	0.089	0.201	0.143	0.088	0.077	0.0396
∗ Bi-GRU w/ Attention	0.230	0.56	0.183	0.235	0.183	0.161	0.130
∗ Zero-Shot GPT-2	0.120	0.035	0.154	0.061	0.032	0.030	0.014
∗ Our Method	**0.340**	**0.156**	**0.308**	**0.321**	**0.219**	**0.187**	**0.108**
	ICMLA						
PREFIX	0.211	0.081	0.185	0.150	0.090	0.075	0.036
TextRank	0.151	0.042	0.139	0.094	0.048	0.038	0.014
LexRank	0.202	0.074	0.179	0.136	0.079	0.067	0.032
∗ Bi-GRU w/ Attention	0.198	0.075	0.161	0.179	0.181	0.158	0.126
∗ Zero-Shot GPT-2	0.115	0.029	0.145	0.063	0.034	0.028	0.013
∗ Our Method	**0.404**	**0.189**	**0.356**	**0.395**	**0.271**	**0.216**	**0.123**

TABLE VII
ROUGE AND BLEU SCORES FOR ALL THREE DATASETS

abstract as the title.

TextRank In this algorithm, Mihalcea and Tarau [21] propose a graph based representation on input text, where sentences are modeled as nodes in a graph with defined weighting scheme. PageRank algorithm is then applied to score each node in the graph and the highest scoring sentence is picked as the title of the given abstract. We use Python package Gensim [5] for implementation purposes.

LexRank In this algorithm, Erkan and Radev [22] propose an unsupervised approach to text summarisation based on graph-based centrality scoring of sentences. We pick the highest scoring sentence as the title of the given abstract. We use Python package Sumy [6] for implementation purposes.

Bi-GRU with Attention We use GRU based encoder-decoder architecture (Sequence to Sequence Network) with attention to produce titles for the given abstracts. We use teacher forcing parameter/ratio [23] to decide on the balance between feeding ground truth compared to treating previous output as input for

[5]Gensim - https://tinyurl.com/y8mg4932

[6]Sumy - https://tinyurl.com/y9zym6sc

Abstract: This paper presents a novel approach for multi-lingual sentiment classification in short texts. This is a challenging task as the amount of training data in languages other than English is very limited. Previously proposed multi-lingual approaches typically require to establish a correspondence to English for which powerful classifiers are already available. In contrast, our method does not require such supervision. We leverage large amounts of weakly-supervised data in various languages to train a multi-layer convolutional network and demonstrate the importance of using pre-training of such networks. We thoroughly evaluate our approach on various multi-lingual datasets, including the recent SemEval-2016 sentiment prediction benchmark (Task 4), where we achieved state-of-the-art performance. We also compare the performance of our model trained individually for each language to a variant trained for all languages at once. We show that the latter model reaches slightly worse - but still acceptable - performance when compared to the single language model, while benefiting from better generalization properties across languages. **Actual**: Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification **Selected Title**: Multi-lingual Sentiment Classification with Weakly-Supervised Data in Short Texts / **RS: 0.11** **Refined Title**: Multi-lingual Sentiment Classification with Weakly-Supervised Data / **RS: 0.05**
Abstract: In this paper, we propose an efficient semantic segmentation framework for indoor scenes, tailored to the application on a mobile robot. Semantic segmentation can help robots to gain a reasonable understanding of their environment, but to reach this goal, the algorithms not only need to be accurate, but also fast and robust. Therefore, we developed an optimized 3D point cloud processing framework based on a Randomized Decision Forest, achieving competitive results at sufficiently high frame rates. We evaluate the capabilities of our method on the popular NYU depth dataset and our own data and demonstrate its feasibility by deploying it on a mobile service robot, for which we could optimize an object search procedure using our results. **Actual**: Find my mug: Efficient object search with a mobile robot using semantic segmentation **Selected Title**: Fast and Accurate Semantic Segmentation on Mobile Robot / **RS: 0.13** **Refined Title**: Robotic Semantic Segmentation with Deep Learning / **RS: 0.00001**
Abstract: This paper proposes an online transfer framework to capture the interaction among agents and shows that current transfer learning in reinforcement learning is a special case of online transfer. Furthermore, this paper re-characterizes existing agents-teaching-agents methods as online transfer and analyze one such teaching method in three ways. First, the convergence of Q-learning and Sarsa with tabular representation with a finite budget is proven. Second, the convergence of Q-learning and Sarsa with linear function approximation is established. Third, the we show the asymptotic performance cannot be hurt through teaching. Additionally, all theoretical results are empirically validated. **Actual**: Online Transfer Learning in Reinforcement Learning Domains **Selected Title**: Online Transfer in Reinforcement Learning: A New Approach / **RS: 0.09** **Refined Title**: Transfer Learning for Reinforcement Learning / **RS: 0.10**
Abstract: We first discuss certain problems with the classical probabilistic approach for assessing forensic evidence, in particular its inability to distinguish between lack of belief and disbelief, and its inability to model complete ignorance within a given population. We then discuss Shafer belief functions, a generalization of probability distributions, which can deal with both these objections. We use a calculus of belief functions which does not use the much criticized Dempster rule of combination, but only the very natural Dempster-Shafer conditioning. We then apply this calculus to some classical forensic problems like the various island problems and the problem of parental identification. If we impose no prior knowledge apart from assuming that the culprit or parent belongs to a given population (something which is possible in our setting), then our answers differ from the classical ones when uniform or other priors are imposed. We can actually retrieve the classical answers by imposing the relevant priors, so our setup can and should be interpreted as a generalization of the classical methodology, allowing more flexibility. We show how our calculus can be used to develop an analogue of Bayes' rule, with belief functions instead of classical probabilities. We also discuss consequences of our theory for legal practice. **Actual**: Assessing forensic evidence by computing belief functions **Selected Title**: Probabilistic Functions for Forensic Evidence / **RS: 0.045** **Refined Title**: Probabilistic and Belief Functions for Forensic Evidence / **RS: 0.095**

TABLE VIII
SAMPLE OUTPUTS FROM SELECTION AND REFINEMENT MODULE FOR GIVEN INPUT ABSTRACT. HERE RS IS THE RELEVANCY SCORE

current step. During training, we feed the generated words instead of actual words as mentioned in [24], we do this specifically 30% of the time. All the parameters were selected based on the best performing parameters on the validation set.

Zero-shot evaluation with GPT-2 We use medium sized GPT-2 pre-trained language model in this case. To induce the summarisation behaviour we add the keyword $TL;DR$: at the end of an input sequence [25].

D. Evaluation and Results

We evaluate our approach by comparing it to the baseline algorithms as described above. We use ROUGE [26] and BLEU [27] metrics for comparison since these are the current standard evaluation metrics in summarisation systems. These metrics essentially depend on n-gram overlaps between a generated text and the actual text with ROUGE measuring recall and BLEU measuring precision.

Table VII shows the result for our automatic evaluation on arXiv test split, ACL and ICMLA datasets. We can see that our approach has high ROUGE and BLEU scores in all the three datasets as compared to the baseline algorithms.

Since, our approach outputs a pair of titles with relevancy scores, we also perform human evaluation to help validate how well does the human selection correlate with the fully automatic pipeline deployment. We did not create separate human evaluation for validating the quality of the titles since the user could select either/neither option if he did not agree with any of the title options provided.

For evaluation, we had 22 human evaluators consisting of Graduates, Post Graduates and PhD students well versed in the technical domain that represented our dataset. Evaluators were presented with a sample of 25 abstracts along with the options of the titles from Selection and Refinement modules and the option of choosing "either/neither".

The results of this evaluation are presented as a Confusion Matrix in the Table V. We can see that precision and recall are 0.76 and 0.86 for Selection and 0.75 and 0.60 for Refinement respectively. Table VIII shows some examples of the our title generation approach which were sampled from the set presented for human evaluation. Examples include abstract, actual, selected and refined titles along with the relevancy

scores. In all these examples, human judgements in-terms of selecting the final title concur with our final selection based on the relevancy scores. Users were not provided with the actual title to avoid any bias while selecting the preferred title.

We also made the following observations for the titles generated by our approach:

- Model correctly reproduces the punctuation separated compound words most of the time. Refer to example 1 in Table VIII.
- Model is able to relate to any specific attributes mentioned in the abstract. As can be seen in example 2 in Table VIII, "Fast and Accurate" are necessary and relevant terms that authors emphasise for the proposed algorithm. Our Generation module was able to generate or reproduce the title with these terms correctly and the Selection module was able to grasp this and select the title.
- Our pipeline preserves the corpus specific phrase structures while producing the titles. In example 3 of Table VIII, we observe that Refinement module was able to correctly re-create the phrase "Transfer Learning" from the partial phrase of the selected title.

V. Conclusion

In this paper, we presented a novel approach for automatic title generation from short text using the pre-trained Transformer Language Model. Our model could produce syntactic and semantically correct titles without the need of using larger dataset for training. We validated our approach on multiple datasets with human and objective evaluation and the results obtained were encouraging. This approach is easily adaptable to different domains given enough data.

The possible future direction for our work is to generate good and effective abstracts automatically from very long documents. This would help create a full pipeline from reading long documents to generating short summary and creating the title for the same.

Acknowledgement

We thank the Center of Excellence on Cognitive Computing, funded by Mphasis F1 Foundation for funding this research. We also thank Dr Prasad Ram and Gooru team (https://gooru.org) for the topical discussions and encouragement.

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[2] R. Jin and A. G. Hauptmann, "Automatic title generation for spoken broadcast news," in *Proceedings of the first international conference on Human language technology research*. Association for Computational Linguistics, 2001, pp. 1–3.

[3] L. Cédric, V. Prince, and M. Roche, "How to title electronic documents using text mining techniques," *IJCISIM*, vol. 4, pp. 2150–7988, 12 2012.

[4] J. Tan, X. Wan, and J. Xiao, "From neural sentence summarization to headline generation: A coarse-to-fine approach." in *IJCAI*, 2017, pp. 4109–4115.

[5] C. A. Colmenares, M. Litvak, A. Mantrach, and F. Silvestri, "Heads: Headline generation as sequence prediction using an abstract feature-rich space," 2015.

[6] J. W. G. Putra and M. L. Khodra, "Automatic title generation in scientific articles for authorship assistance: a summarization approach," *Journal of ICT Research and Applications*, vol. 11, no. 3, pp. 253–267, 2017.

[7] X. Gu, Y. Mao, J. Han, J. Liu, Y. Wu, C. Yu, D. Finnie, H. Yu, J. Zhai, and N. Zukoski, "Generating representative headlines for news stories," in *Proceedings of The Web Conference 2020*, 2020, pp. 1773–1784.

[8] K. Murao, K. Kobayashi, H. Kobayashi, T. Yatsuka, T. Masuyama, T. Higurashi, and Y. Tabuchi, "A case study on neural headline generation for editing support," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, 2019, pp. 73–82.

[9] K. Lopyrev, "Generating news headlines with recurrent neural networks," *arXiv preprint arXiv:1512.01712*, 2015.

[10] D. Gavrilov, P. Kalaidin, and V. Malykh, "Self-attentive model for headline generation," in *European Conference on Information Retrieval*. Springer, 2019, pp. 87–93.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv preprint arXiv:1807.03819*, 2018.

[13] Z. Chen, H. Eavani, W. Chen, Y. Liu, and W. Y. Wang, "Few-shot nlg with pre-trained language model," *arXiv preprint arXiv:1904.09521*, 2019.

[14] N. F. Rajani, R. Zhang, Y. C. Tan, S. Zheng, J. Weiss, A. Vyas, A. Gupta, C. XIong, R. Socher, and D. Radev, "Esprit: Explaining solutions to physical reasoning tasks," *arXiv preprint arXiv:2005.00730*, 2020.

[15] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[16] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[17] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[18] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[19] Q. Wang, Z. Zhou, L. Huang, S. Whitehead, B. Zhang, H. Ji, and K. Knight, "Paper abstract writing through editing mechanism," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 260–265. [Online]. Available: https://www.aclweb.org/anthology/P18-2042

[20] D. Vallejo-Huanga, P. Morillo, and C. Ferri, "A dataset of attributes from papers of a machine learning conference," *Data in brief*, vol. 24, p. 103836, 2019.

[21] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[22] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[24] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[25] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[26] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://www.aclweb.org/anthology/P02-1040