Applying transfer learning to abstractive summarization

submitted in partial fulfillment for the degree of master of science

Jeffrey Tsang
11157402

master information studies
data science
faculty of science
university of amsterdam

July 1, 2019

	Internal Supervisor	External Supervisor
Title, Name	dr. Pengjie Ren	Ruben Gort
Email	p.ren@uva.nl	ruben.gort@accenture.com
Affiliation	University of Amsterdam	Accenture

UvA

Applying transfer learning through pre-training of public data sets to abstractive summarization

Jeffrey Tsang
jeffrey.tsang@student.uva.nl
11157402
Universiteit van Amsterdam

ABSTRACT

Abstractive text summarization is a Natural Language Processing (NLP) task of generating summaries that outlines all core topics within a document in such a way that mimics a human-written summary. Recent developments in deep learning has increased the feasibility of abstractive text summarization models. However, a large bottleneck is the amount of labeled data needed to train such models. A commonly used technique for coping with this issue is through the use of transfer learning, where knowledge is transferred from a source task to a target task for transfer learning. In this paper, we explore the concept of transfer learning by using the transfer learning method of pre-training. Three large corpora are used for pre-training the model, and a smaller corpus of blogs of Accenture is used as the target task. In addition, two different learning objectives are used for pre-training; summarization and language model. In general, we found that the performance improved after pre-training when evaluated with the ROUGE-metric. The learning objective of language modeling seemed relatively ineffective in the experiment we conducted. On the other hand, we were able to generate acceptable summaries while pre-training it with the learning objective of summarization that clearly outlines some core concepts of the article. The summaries were not without issues, with some being mediocre in readability, due to some grammatical incorrectness. In conclusion, applying transfer learning to the task of automatic text summarization shows promise, but is not ready to be implemented yet. The selection process of data sets, and more also fine-tuning strategies have to be considered in future studies.

1 INTRODUCTION

In the current age, there is an enormous amount of information that is available and it has become increasingly more important to find and interpret relevant information. Companies are also increasingly more focused on gathering and using data for competitive advantage. With the growing information that is available to a company, text summarization can be a helpful tool for absorbing and interpreting text information. There is an abundance of information that might be usable for companies to support their decision making. Provost and Fawcett [24] describe that due to the sheer volume and variety of data in the current age, manual analysis of data is no longer a feasible strategy. Especially due to the increasing amount of online publishing and electronic documents, condensing the information within each document might be beneficial to the time and capacity of users to make the most use out of their information [28]. Some other fields where automatic text summarization has been explored are legal documents [9], and medical reports [1]

Automatic text summarization is a technique for condensing text to a short length text that includes all the important information [11]. There are two major approaches to text summarization, extractive and abstractive. Extractive summarization aims at selecting important words and sentences from a document to construct the summary. Abstractive summarization aims to learn the core concepts within a document and generate sentences in natural language, which is more similar to a human-like approach [13]. These abstractive models are capable of producing novel sentences out-of-vocabulary tokens (OOV), including words that are not present in the input vocabulary.

In this thesis, we will adopt an existing model, namely the Get To The Point (GTTP) pointer-generator network as proposed by See et al. [29]. This is primarily an abstractive approach with a small extractive element. It had achieved state-of-the art results for an abstractive approach during its introduction. This model can be seen as a sequence-to-sequence network with added components for dealing primarily with two common problems previous models faced; repetitiveness, and incorrect fact reporting.

One of the main challenge for the task of automatic summarization is the absence of labeled data that can be used for supervised methods, also including the more sophisticated deep learning models that have increased in popularity. In the case of training a summarization model, a large number of documents is needed to train such a model. To add to this, corresponding summaries are often absent in a large portion of human-written documents. One method for coping with this issue in the field of Machine Learning is through transfer learning. Transfer learning is described as extracting knowledge a source task and applying this gained knowledge to a target task [23]. One method for applying transfer learning is by deploying a pre-trained model that is trained on a certain source domain to a different task domain and is already proven to be very effective in the area of computer vision [23, 36]. The intuition behind transfer learning for language processing is that languages often share common semantic structure, and grammar [35]. This research hopes to contribute to the understanding of applying transfer learning to the NLP task of automatic text summarization.

To assess the effectiveness of pre-training, we compare the results of our target task without pre-training against several models where we apply pre-training. The GTTP pointer-generator network is used firstly for pre-training with several large data sets. We selected three large data sets suitable to train such models on; CNN/DM, BBC, and Guardian. We also try two different learning objectives for pre-training; summarization, and language modeling. This results in a total of 6 models. After pre-training each model separately, it was fine-tuned on a small corpus of Accenture blog articles. We evaluate these summaries primarily with the quantitative

ROUGE-metric, but also through visual inspection. The experimental results show that pre-training increases the performance considerably.

This research will contribute to a deeper understanding of applying transferred learning on text applications, specifically the task of abstractive automatic text summarization. The main contributions of this thesis can be summarized as follows:

- The paper shows pre-training on three different corpora for abstractive text summarization models; (1) CNN/DailyMail, (2) Guardian, and (3) BBC.
- This paper explores two different pre-training learning objectives for the task of text summarization; summarization and language-modeling.
- This paper explores fine-tuning pre-trained models on a smaller target domain, and states its limitations.

2 RELATED WORK

A considerable amount of literature has been published on automatic text summarization. This Section will dive deeper automatic text summarization models and their progression throughout the years. Furthermore, there is a subsection for covering literature from transfer learning.

2.1 Automatic Text Summarization

When looking at it from a machine learning approach, the task of text summarization can take the form of both an unsupervised and a supervised problem [12, 37]. With traditional machine learning techniques, meaningful features are extracted from the sentences (e.g. keywords, relevance). Using these features, the task of text summarization can be seen as a binary classification problem. Keyphrases within a document are defined as 'positively' labeled sentences that are included in the summary [10]. Sentences that are not included in the summary are labeled 'negative'. There are various features that are extracted from the sentences, such as the length of a keyphrase or the relative importance of words within a keyphrase through metrics such as TF-IDF [15]. Through the use of the labeled data, the classifier can learn the weight of each feature. The trained model would then receive a document as an input, classify all phrases as 'positive' or 'negative' and including those keyphrases in the summary. Some common approaches of text summarization with a classifier includes Naive Bayes, decision trees, support vector machines, Hidden Markov models and Conditional Random Fields [2]. The latter two often perform better, as the model assume a dependency between sentences. More recently, there are more sophisticated approaches for this task using deep learning methods, which will be the primary focus of this research.

There are two primary approaches to automatic text summarization, namely extractive and abstractive. An extractive approach selects key phrases from a document and rearrange it to a summary, similarly to the human approach of using a marker to highlight salient sentences. An abstractive approach generates novel sentences that underlines all the core concepts within a document [6, 13, 28]. The GTTP model that is adopted in this thesis is a combined abstractive and extractive approach. Below subsections will further describe the advances for both approaches.

2.1.1 Extractive approach. A recent study Cheng and Lapata [4] introduced a neural network based encoder-extractor architecture for generating extractive summaries. The proposed model achieved near state-of-art results and showed the potency of deep learning methods for extractive summarization.

Nallapati et al. [21] proposed a novel training method for extractive such models. The training is done with an abstractive approach, thus only requiring human generated summaries similarly to training abstractive models.

Ren et al. [26] proposed an extractive model reaching state-of-the-art performance. This study offers a comprehensive analysis on the use of sentence relations.

Collectively, these studies outline the feasibility of using extractive approaches. A strong aspect of extractive approach is that the summaries are completely accurate in fact reporting due to the extractive nature. However, abstractive summarization has the asset of being more fluent to read generally.

2.1.2 Abstractive approach. One of the earlier deep learning method for abstractive summarization is the implementation of an Attentional Encoder-Decoder Recurrent Neural Network for the task of text summarization, as proposed by Nallapati et al. [22] . This model was initially proposed for the task of machine translation, but has been successfully applied to text summarization.

Two prominent limitation is that summarizing larger documents may result in a lot of repetitive text, -and some of the information is inaccurately described. This can be explained by that some information is considered important by the model thus it is repeated often. In a key study conducted by See et al. [29], a solution for this limitation was proposed by combining an extractive- and abstractive approach. A pointer mechanism was introduced as an addition to the attentional sequence-to-sequence network to cope with the issue of inaccurate information. This pointer mechanism copies words from the source text to the summary, similarly to an extractive method. As explained previously in Subsection 2.1.1, this has the large benefit of correctness in fact reporting. In addition, coverage was included in the network which was shown to be very effective for reducing repetition.

A recent study by Zhang et al. [39] incorporated a natural language generation model based on BERT [7]. By using a pre-trained language model in the encoder and decoder this model achieved state-of-the art results on ROUGE-1, ROUGE-2 and ROUGE-L. We will attempt something similar through pre-training our model with a language model learning objective.

2.2 Transfer Learning

Allahyari et al. [2] describes that one of the primary bottleneck for supervised learning is the difficulty of obtaining labeled data to train a classifier. In addition, the labeled data used should also match the documents that are used as input for the summarizer. Some approaches have been proposed for coping with this issue. Firstly, semi-supervised learning approaches can be deployed, which requires less labeled data. To determine the effects of using semi-supervised learning, Wong et al. [37] found in their study that using semi-supervised learning, the performance was relatively

close, while the labeling time was reduced by roughly 50%. Secondly, transfer learning can be applied using different data sets. There are multiple annotated corpora publicly available, which are readily available for training classifiers. One such example of an annotated corpus is the BC3 (British Columbia Conversation Corpus)[1], established by Ulrich et al. [34]. These methods however often assume that the labeled data and unlabeled data come from the same distribution, while transfer learning allows the domains and tasks to be different. Similarly to our work, the word distribution is different from the data sets we have.

Due to the increasing adoption of deep neural networks, there have been multiple research done on how to specifically apply transfer learning. Yosinski et al. [38] have explored two different fine-tuning techniques for their neural network; (1) fine-tuning features and (2) freezing features. An important finding was that they found initializing with transferred features can boost generalization of the network. Mou et al. [20] discusses that the use of transfer learning is not widely adopted for NLP tasks unlike other fields such as image processing. NLP tasks faces the same problem of not having sufficient captured data which makes transfer learning all the more desirable. They have found in their paper that it can be applied successfully, often seeing an increase in performance and training time. The same study also describes that the added benefit of transfer learning is coping with the overfitting issue that is often prevalent in deep neural networks [20]. We will consider this same strategy in our work.

3 METHOD

3.1 Task Definition

Given large-scale publicly available summarization data sets and a small target summarization data set, we aim to present a method for pre-training and fine-tuning an abstractive summarization model. Given a source domain D_s with source task T_s, we attempt to transfer the knowledge from pre-training the network on the source domain to a target domain D_t with task T_t.

3.2 GTTP Network

This thesis makes use of the GTTP pointer-generator network as proposed by See et al. [29]. More specifically, a Tensorflow implementation of the GTTP has been used as published by the author. [2]. In this subsection I will describe the model accordingly to its three main components.

Firstly, there is a sequence-to-sequence model that serve as their baseline model as also depicted in Figure 1. The sequence-to-sequence model consists of an encoder, and a decoder both of which are composed of LSTM (Long short-term Memory) units and an attention mechanism. LSTM is a special type of RNN (Recurrent Neural Network) that deals with the vanishing gradient problem that occurs often in an RNN. [31]. The vanishing gradient problem describes the common effect that the gradient that is propagated gets gradually smaller, and thus no longer has any real effect on the weights the further it goes back in the network [3]. This can be problematic as new tokens can be correlated with tokens from a much earlier time-step, also called long-term dependency. Simply

[1] 'https://www.cs.ubc.ca/ rjoty/Webpage/resources.htm'
[2] https://github.com/abisee/pointer-generator

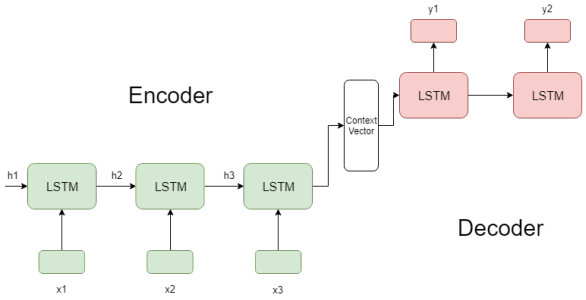

Figure 1: A simple one-layer sequence-to-sequence network

put, LSTM's solve the vanishing gradient problem and also allows capturing long-term dependencies in sequential data. Given a text as input with tokens $\{x_1, x_2, ..., x_T\}$, the encoder part of the model takes each token sequentially as input. At each time-step, the input to the encoder LSTM is both the current token x_t , as well as the hidden state from the previous time-step, $h_t - 1$, where $_t$ stands for timestep and h for hidden state. In addition, LSTM in contrast to regular RNN, also encode a memory cell state c_t. The hidden state is at a given LSTM cell is defined by the following function:

$$(h_t, c_t) = (x_t, (h_t - 1, c_t - 1))$$

Once the final token has been passed into the encoder, the final hidden state hT is defined as the context vector z. This vector captures the entire source text and is used as input for the decoder. Similar to the encoder, the decoder takes the hidden state s_t during each timestep, and also the word y_t. Using the hidden state s_t, a predicted token $\hat{y}_t$ is produced by a soft-max layer over the target vocabulary. The formula to calculated the hidden state is as follows (cell state c_t is the same as the encoder part)

$$(s_t, c_t) = (y_t, (s_t - 1, c_t - 1))$$

Secondly, there is a pointer mechanism incorporated in this model that directly copies word from the source text. This component is an extractive approach, that is included in the GTTP. The core idea is to improve accuracy of factual reporting, and also to better handle words that are OOV (Out of Vocabulary). During each time-step t of decoding, the pointer has has a generation $P_gen \in [0, 1]$ that determines whether to generate a word from the vocabulary or copy it from the source text.

Thirdly, coverage is included in the model, an idea borrowed from Tu et al. [33]. The author proposed a coverage vector for keeping track of attention history, primarily with the goal of reducing repetition. In the GTTP implementation, a coverage vector consists of the sum of attention distribution of the decoder.

Figure 2 shows the GTTP network in its whole. Concludingly, This pointer generator network can be seen as combining the strengths of extractive and abstractive approaches. The pointer allows for extractively copying words from the source text. This has the benefit of increasing the accuracy of factual details. In the case of words that are less occurring but still very important like certain names or numbers, the pointer can still input the OOV word. An additional benefit is the relatively smaller vocabulary

size needed to train the model, which speeds up the training by a considerable time.

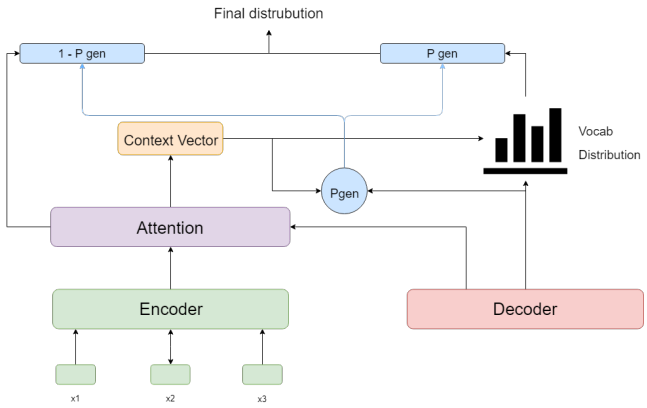

Figure 2: A pointer-gen network

3.3 Research methods

3.3.1 Overview. The primary task of this research is summarizing a small target corpus of blog articles from Accenture using various transfer learning techniques. This is subsequently done with three subtasks; (1) pre-training on three different data sets, (2) pre-training it on different tasks, using the same data sets, (3) finetuning it on a smaller target corpus, and lastly (4) evaluating the models with ROUGE. Figure 3 depicts the workflow for these three tasks.

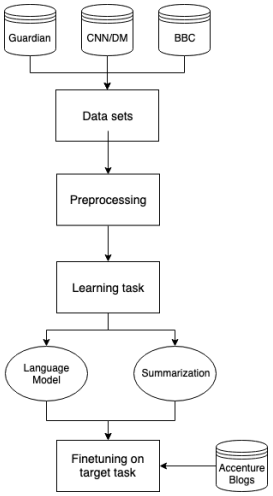

Figure 3: Workflow of the three tasks

(1) Using the large data sets as source domain, we will first pre-train the models on them. The three data sets are different stylistically, as the CNN/DM contains news articles. The guardian & BBC corpora in turn contain live blogs, which is substantially different as the source text is often not as coherent but rather loosely related sentences. The three data sets used are thoroughly described in Section 4.1.

(2) The first pre-training objective we will perform is summarization. Consistent across all three data sets each training example contains the article itself, and a summary as label. The summary contains of multiple sentences, each sentence marked with a @highlights as label. The network is trained to predict a partial summary label given a partial article as input. Details are given in Section 4. As discussed above, the network is pre-trained to generate a summary with the input of the source domain articles. The articles in the target task, however, do not clearly have a human written summary. We had made an assumption that the first paragraph can be perceived as the summary of the corresponding article, since it is always used as the article preview. Since there is no guarantee, for example, people using the preview text merely to get attention, we solved this by manually checking each data example from the test set. There may still be some noise in the training set however.

The second pre-training task is language modeling. Some recent work has suggested that training a language model might be applied to a large number of NLP tasks, including summarization [7, 16]. We use probabilistic language modeling where the goal is to calculate the probabilities of the sequence of words [30]. Here each sentence in the article is split into two parts. For example, the sentence "The weather is sunny today" would be split into the "the weather is" and the target "sunny today". The objective is to predict subsequent words given a sequence of words.

(3) Using the pre-trained models from (1) and (2), we will finetune the network on the target task. This will be done by retraining the network without re-initializing the parameters, in other words keeping the learned weights.

(4) Finally, the performance of the models across all experiments will be evaluated and compared using the evaluation metric ROUGE, which will be further elaborated on in Subsection 4.3.

3.3.2 Pre-training based transfer learning. In this research, transfer learning is used to capture linguistic phenomena from the source data and use this knowledge on the target data set of Accenture blog articles. As shown in Figure 4, transfer learning attempts to transfer the knowledge from a source domain to a smaller target domain. Transfer learning has two domains; a source domain D_s, and a target domain D_t. A domain consists of two components; A feature space χ and a marginal probability distribution $P(X)$, where X is a data point from the collection $X = \{x1, x2, ... xn\} \in \chi$. In addition, a domain consists of a task D_T that exists of two components; a label space Y, and a predictive function f with input a given pair of input feature vector and label (x_i, y_i). To sum it up, transfer learning aims to learn the distribution $P(Y_T|X_T)$ in a given domain D_T with the knowledge gained from D_S and T_S.

In our setting, we first train the parameters of the model on the source domain separately. Given any data set we use, we have a different source domain D_S. Here the features, the vectorized words in this case, can differ substantially across source and target domain. While there are common words that should be domain independent and should occur roughly at a comparable rate, there are also more domain specific words that are tied to the type of documents present in such domain. For example, the central overarching topic in our target domain is innovation, where some technology terms such

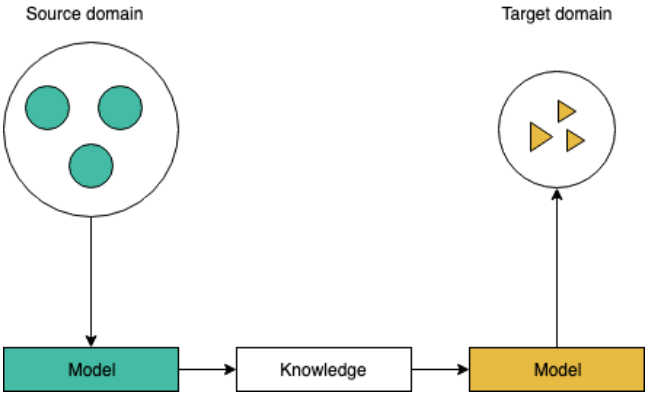

Figure 4: Transfer Learning

as "SAP" or "DMBS" are common which may not be the case for the more general documents of our source domain. The marginal distribution is thus different, so $P(Y_S|X_S) \neq P(Y_T|X_T)$, also called the frequency feature bias [36].

3.3.3 Finetuning. With the knowledge gained from pre-training, we will fine-tune it to the target task. Fine-tuning is described as using parameters from an existing worked that is trained on a related data-rich task, and optimize the parameters accordingly to the new target task [17]. The learned weights from pre-training of the network will thus be used as the starting point for our retraining. During retraining all the weights will be updated through back-propagation. Yosinski et al. [38] reports that fine-tuning without freezing layers often leads to better generalization, at the expense of longer computation time. Fine-tuning all the parameters without freezing any transferable layers is also the approach we will use.

4 EXPERIMENTAL SETUP

In this section the experimental setup is described with the goal of answering the research questions of this thesis. Firstly, the research questions are given in Section 4.1. Secondly, the data set is described in Section 4.2. Thirdly, we describe the evaluation metrics to be used for assessing the models in Section 4.3. Fourthly, we show the comparison between the different experiments performance-wise in Section 4.4. Lastly, we present the technical details of the model and training in Section 4.5.

4.1 Research Questions

We aim to seek out whether applying transfer learning through using pre-trained methods might be helpful for training models with a small target domain. For this, we formulated the following research question.

(RQ1) "Does pre-training a model on publicly available annotated corpora improve the performance level for the task of abstractive text summarization given a small target domain? "

We use three large public data sets for pre-training the models. As the data sets differ in characteristics such as size, topics, we formulated the following question:

(RQ2) "Which of the public data sets gives the best performance level?"

We perform pre-training with two learning method; namely summarization and language modeling. This leads to the following question

(RQ3) "What pre-training objective performs the best between summarization and language modeling for the task of text summarization? "

4.2 Data

Four data sets are used in this paper, three for pre-training and one as the target domain. Each will be further described in the following subsections.

4.2.1 Cnn/Dailymail Corpus. The cnn/dailymail corpus, as proposed by Hermann et al. [14], is used for pre-training the model. This corpus is used in many other literature [21, 22, 29]. It contains news articles from news outlets CNN and Dailymail and their corresponding summaries amounting 312,085 pairs. An 80-10-10 split has been done consequently on the data set resulting in a training- (287,227 pairs), validation- (13,368) and test split (11,490). The data set was retrieved with source code found on Github. [3].

4.2.2 BBC & Guardian Corpus. The BBC corpus, used for pre-training the model, was proposed by P.V.S. et al. [25]. These articles are live-blogs that are dynamically updated in a given timeframe, and a summary is written by a journalist when the broadcasting is ended. The BBC corpus in contain 10,537 training pairs, 77 validation pairs, and 75 test pairs. The data set was obtained directly from the Author but can also be retrieved with their source code found on Github. [4]

4.2.3 Guardian. Similar to the BBC corpus, the Guardian corpus was proposed by P.V.S. et al. [25]. The same retrieval method was used for this corpus. The Guardian corpus contains of live-blog articles and their human-written summaries amounting 9655 pairs. A split has been done consequently on the data set resulting in a training- (9322 pairs), validation- (167) and test split (166).

4.2.4 Accenture Blog Articles. The articles from Accenture are written as blogs and retrieved from the Accenture-Insights portal via a scraper I built. [5] Multiple domains were scraped; Netherlands, Belgium, Luxembourg, and United States and only articles written in English was scraped. In addition, while all these articles focus on innovation, there are numerous topics that are covered (e.g. cloud computing, human resource management, artificial intelligence etc.). The total corpus amounts to 950 pairs. A 80-10-10 split has been done consequently on the data set resulting in a training- (760 pairs), validation- (95) and test split (95). This data set serves as our target domain.

[3] https://github.com/hpzhao/non-anonymized-CNN-DailyMail
[4] https://github.com/UKPLab/lrec2018-live-blog-corpus
[5] https://github.com/Jeblii/MSc-Thesis-Jeffrey-Tsang-UvA-Accenture/tree/master/Crawler

Learning Objective	ROUGE-1			ROUGE-2			ROUGE-L		
	Precision	Recall	F1 score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Summarization	3.97	3.02	3.15	0	0	0	3.97	3.02	3.15
Language Modeling	7.66	4.59	5.44	0	0	0	7.66	4.59	5.44

Table 1: ROUGE-score for training only on the target taskwithout pre-training

4.3 Evaluation Metrics

Traditionally, the quality of generated summaries were judged and evaluated by human judges on its usefulness and reliability [19]. This is a very time-consuming process, leading to multiple proposals for automatically evaluating summarizations. Lin [18] proposed one of the most popular quantitative evaluation, the ROUGE-metric, that is currently the most commonly used metrics for assessing artificially produced summaries. Generally speaking, this metric assesses the completeness, but also the correctness of the summary. ROUGE has different variations, each with different metrics, for measuring the quality of a summary. Some examples of the variations are ROUGE-N and ROUGE-L, correspondingly comparing the n-gram (e.g. unigram, bigram) and longest sequence of words overlap between the golden standard and model generated summaries. The ROUGE score can however be faulty, for example a good score in ROUGE-1 does not necessarily mean that the summary is readable or efficient in presenting only the salient information.

Similar to many other works, we will use the F1 variant of ROUGE for comparing the performance of the models [5, 21, 22, 29]. In addition, we will also report the precision and recall for the ROUGE metric. Three variants of ROUGE will be used for evaluation. Firstly, ROUGE-1 which measures the overlap of uni-gram, thus single words. Secondly, ROUGE-2 which measures the overlap of bi-grams. Lastly, ROUGE-L which measures the longest overlapping sequence. Below is shown how these metrics are calculated for ROUGE-N.

$$Recall = \frac{\text{number of overlapping n-gram}}{\text{total n-gram ref summary}}$$

$$Precision = \frac{\text{number of overlapping n-gram}}{\text{total n-gram generated summary}}$$

4.4 Comparative baseline performance

We use the same models across all the different experiments, and do not have a separate baseline model. Instead, the baseline performance is defined as only training the model on the corpus of the target task. Table 1 presents the ROUGE scores of the results that are generated from this trained model. We use ROUGE-1, ROUGE-2, and ROUGE-L as metric to the performance of the models.

Figure 5 shows an example of a model generated summary in comparison to a reference summary. As can be seen, the output is not a readable summary, and contain of a lot of [UNK] tokens meaning that the word is not contained in the vocabulary set. Overall, the baseline performance is rather poor. Some other common problems we inspected visually were the inability to deal with infrequent words, a lot of repetition and also reporting incorrect factual details. The objective of the experiments is to improve upon the performance.

Original Article (truncated): Imagine calling for a taxi every Thursday at around 19.00 hours. Then one day, a cab arrives in your street uninvited. The driver doesn't make his presence known, but simply waits for you to request a ride, at which point he drives up to your house and says: "I had been expecting your call, so I showed up in advance. How's that for customer service?" It would be a rather unsettling experience, wouldn´t it? Creepy as this sounds, this is exactly what is happening online. Data-driven machines have all the technical know-how to provide personalized marketing experiences, but they lack the natural boundaries we apply in day-to-day human contact. This is perfectly illustrated by the well-known example of an online retailer who figured out a teen girl was pregnant before her father did. Although this is a fairly old case, not much has changed since then: by tracking all and everything, we intrude in the lives of our customers without truly taking the time to get to know them, or to build up trust. It appears we also lack relevance, as we blatantly bombard our target audience with hotel ads even after they have booked.
Is our message to refrain from data collection? To stop creating personalized marketing and service experiences? Absolutely not. In fact, over the past decades, we evolved from highlighting quality ("Made in America!") to emphasizing image (Apple springs to mind) to offering personalized experiences.

Reference (golden) summary: unrestrained data collection won't per se lead to relevant customer experiences. in fact , when it comes to collecting data , less is more. this is the first of two articles on creating relevant data-driven customer experiences by gathering indiscriminate data and starting relevant customer dialogues .

Model generated summary: [UNK] , [UNK] [UNK] , [UNK] [UNK] , [UNK] [UNK] , [UNK] [UNK] [UNK] [UNK] [UNK] , [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] .

Figure 5: An example article from the target task, along with the reference summary, and model generated summary

4.5 Technical Details

The GTTP model was used in all the experiments that are described in the previous Sections. The model was initiliaized with 256 hidden-dimensions, and 128-dimensional word-embeddings. The vocabulary used for all experiments was constructed from all the data sets, and limited to 50.000 entries, similarly to the work of See et al. [29] who argues that the vocabulary size is sufficient given the pointer's ability to generate words that are OOV. The network has 21,500,753 trainable parameters. For all the experiments, we trained using across two machines; one with a GTX1060, and a virtual machine on AWS with a Nvidia K80.

4.5.1 Summarization Task. Pre-training the model was done for around 15 epochs which took a couple of days, depending on the data set used. Training was first done on a truncated article of 200 tokens and the decoder steps was narrowed down to 50 tokens. This was done to speed up the training, as it has little effect the final performance of the model [29]. During training, we gradually increased the size to 400 tokens for the article, and 100 tokens for the generated summary. A batch size of 16 was used for the training consistently during each step. We started with an initial learning rate of 0.15 in combination with the optimizer Adagrad which is an adaptive gradient algorithm that that adaptively scales the learning rate for different parameters [8].

For fine-tuning the model on the target task of Accenture blog articles, we retrained the pre-trained models for 10 epochs. This was done with a truncated article of 400 tokens and a summary of 200 tokens. We did not first reduce these sizes during training like the previous steps as the corpus was much smaller and thus relatively fast to train.

4.5.2 Language Model Task. For pre-training the model on the learning objective of language modeling I trained around 15 epochs. Since the training examples are made out of only a sentence, we

did not truncate the tokens like for the summarization task. For the exact same reason, the batch size was increased to 128. Similar to the summarization task experiment, we start of with the initial learning rate of 0.15, using the same Adagrad optimizer.

Fine-tuning the model on the target task was done by retraining for an additional 10 epochs on the target task and similarly done without truncation.

5 RESULTS

In this Section, we present the results of the experiments as described in Section 4. Table 2 presents the results obtained from pre-training it on both summarization and language model task. The results are derived using the Python package pyrouge [6] a wrapper that uses the official Perl ROUGE implementation. The objective of these results was to answer RQ1, where we aim to explore the performance of applying transfer learning to the task of text summarization. As a comparative baseline we had first trained the model without any pre-training with the results shown in Table 1.

As can be seen from this learning objective, the performance has already improved from the baseline performance. When training with the learning objective of summarization, it can be seen that the data sets are very similar in their performance. The Guardian data set performs slightly worse in comparison to the other two data sets of BBC and CNN/DM. The BBC data set has the highest ROUGE-1, ROUGE-2, and ROUGE-L scores with one exception. The ROUGE-1 precision of the CNN/DM pre-training is scarcely any higher with 0.23 points. Interestingly however, for all three pre-training methods, the ROUGE-2 score is very low when compared to the other metrics.

In addition to the ROUGE score, we present two examples of summaries. The first is a summary of which we consider relatively good and is presented in Figure 6. While it can be argued that the sentences are not the most fluent, they are grammatically correct for the most part. This example also shows that the generated summary captures some salient information that is not included in the reference summary, like mentioning how security systems can have an impact on the global supply chain. We also present a poorly generated summary, as depicted in Figure 7. For the badly generated summary it can be seen that there are several limitations. Firstly, it struggles with out of vocabulary word, as seen with the [UNK] token. Secondly, the summary consists of a lot of repetitive sentences as the phrase "it 's not a choice" is repeated two times. Concludingly, this summary neither captures the salient information, nor is it readable.

When training with the learning objective of language model, it can be seen that all three data sets perform poor. The best ROUGE-1, ROUGE-2, and ROUGE-L score does not give a clear improvement over the baseline performance, and in the case of BBC it is even worse.

Upon visual inspection of the generated summaries, it can be clearly seen that the method we had used does not generate readable summaries. The results are similar to the ones generated without pre-training shown in Figure 7.

A comparison between the two results in Table 2 show that pre-training on a summarization task performs considerably better across all three ROUGE metrics compared against pre-training it on a language model objective.

[6] https://pypi.org/project/pyrouge/

Original Article (truncated): Let 's be clear , the industry 's digital transformation comes with massive benefits .On mine sites across Australia , autonomous vehicles , haulage and drilling systems are improving workplace safety and creating production efficiencies But , with an increasing number of connected devices and systems , connected mine sites are more vulnerable than ever before to cyber security breaches and attacks from criminals , hacktivists , competitors and other nation states. And you don't have to be an evil genius to bring a mine to a halt .
An employee inadvertently clicking on a malware email can now compromise critical systems , meaning the whole mine has to be shut down while the issue is resolved. Even an hour of unplanned down time will have cascading effects across a global supply chain .
So, how can mines build cyber resilience and grow with confidence ? You need to evaluate your existing vulnerabilities and build in security from the outset . Above all , this means taking an end-to-end view rather than a siloed view of potential threats. In other words , define the `` service '' being provided as the mine 's primary asset -- and IT networks and equipment as support structures rather than stand-alone entities .

Reference (golden) summary: back in the day , cyber security in mining was limited to protecting critical it systems from natural disasters or physical tampering .
today , with a connected work environment , a mine 's `` attack surface '' has increased exponentially -- leading to a new raft of nightmare scenarios for mine executives to contend with .

Model generated summary: proactive, the industry is while the industry is happening to a new era of potential threats. but with many an eye new security systems, we can have an impact on the global supply chain.

Figure 6: An example of a good summary along with; article, reference summary, and model generated summary

Original Article (truncated): To drive business value faster , many companies need to leverage a managed service provider -LRB- MSP -RRB- , which can not only migrate you to the public cloud , but then optimize performance and drive innovation and growth over time . We 're proud at Accenture that we 've achieved the highest and furthest overall position for our `` ability to execute '' and `` completeness of vision '' in a new report from market research and advisory firm Gartner , Inc. . Public Cloud Infrastructure Managed Service Providers , Worldwide Magic Quadrant 2018 . The report analyzed 20 vendors that offer managed services and professional services related to infrastructure and platform operations for one or more hyperscale integrated Infrastructure-as-a-Service -LRB- IaaS -RRB- and Platform-as-a-Service -LRB- PaaS -RRB- providers -- specifically , Amazon Web Services , Microsoft Azure and Google Cloud Platform . To drive business value faster , many companies need to leverage a managed service provider -LRB- MSP -RRB- Why would a company want to seek out a managed service provider for public cloud services ?

Reference (golden) summary: more and more companies are moving portions of their it estate to the public cloud . why ? companies usually cite some common reasons : to keep infrastructure costs low , gain greater elasticity and shift capex spending to opex .

Model generated summary: it 's not a choice .
it is not a choice .
it [UNK] that is not a choice .
it just about my [UNK] companies , drive business transformation , it is not just about any length stakes and that is not long those making to someone the cloud means are long gone is more so far .

Figure 7: An example of a poor summary along with; article, reference summary, and model generated summary

6 ANALYSIS

In this Section we further elaborate on the results by performing an analysis. We will structure it based on RQ2 and RQ3.

Data set	ROUGE-1			ROUGE-2			ROUGE-L		
Learning with summarization task									
	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score
CNN/DM	20.19	26.31	21.27	2.58	3.42	2.76	16.72	22.11	17.76
Guardian	18.22	23.77	19.29	1.2	1.77	1.36	15.16	19.85	16.05
BBC	20.26	26.08	21.32	1.84	2.55	1.98	17.11	22.23	18.09
Learning with language modeling									
CNN/DM	8.88	9.41	8.65	0.5	0.4	0.6	8.74	9.20	8.48
Guardian	7.64	5.64	6.14	0.31	0.17	0.21	7.38	5.55	6.0
BBC	5.36	4.24	4.80	0.12	0.05	0.09	5.76	4.85	5.12

Table 2: ROUGE-score for training only on the target task without pre-training

6.1 Performance across data sets (RQ2)

The two best performing data sets for pre-training, based on the ROUGE-score was the BBC data set, and the CNN/DM dataset. Interestingly however, is that the BBC data set is a much smaller corpus, around 30 times (roughly 290,000 articles). Furthermore, due to the much smaller size the pre-training took much fewer iterations and as a result much quicker to train. Similarly, the BBC data set did not differ substantially in performance when compared CNN/DM. We can think of possible explanations for these results. Firstly, the relatively good results from pre-training it on the BBC data set could be related to the relatedness of the source and target data set. Some other authors have stated that the more related the target task is to the source task, the easier it is to apply transfer learning on [27, 32]. Secondly, it might be the case that the trade-off between training time and performance gets gradually reduced with more data at a certain point. However, this explanation should be interpreted with caution, as we did not conduct an experiment to test this hypothesis.

While similar in size to BBC, the Guardian data set performed the worse across the ROUGE metrics we use. P.V.S. et al. [25] reports that the Guardian corpus covers more topics, but has relatively fewer documents per topic. It may be possible that due to the smaller number of documents covering each topic, the model does not gain sufficient knowledge. Another possible explanation is that the topics present in the BBC corpus are more closely related to the topics within our Accenture blog target domain.

6.2 Different learning objective (RQ3)

To apply transfer learning to the task of text summarization, we first pre-trained three data sets. We used two different learning objectives, namely summarization and language model and trained the model separately. This resulted in six different models that were afterwards fine-tuned on a smaller target data set consisting of blog articles. Below we will discuss the results of both learning objectives. Generally, the learning objective of summarization for this task is much more effective than training the model on language modeling.

6.2.1 Summarization Task. One of the issues in some of the summaries was the occurrence of [UNK] tokens. A large portion of generated summaries contained no [UNK] tokens, some contained a few, and some summaries contained multiple [UNK] tokens, sometimes in a sequence. The result is a summary that is unreadable. We used a smaller vocabulary for training, as this quickens the training time, and the pointer should be able to deal with OOV words [29]. A possible explanation is that a larger vocabulary size had to be used for coping with this issue.

It can also be argued that the ROUGE metric does not always give a the most accurate depiction. We performed visual inspection across all generated summaries to compare the two best performing data sets further; BBC and CNN/DM. While the BBC scores somewhat higher purely due to more overlapping words, we found that the summaries of CNN/DM were in some cases much more fluent in readability.

6.2.2 Language Model Task. We see that the language model gets considerable lower performance then the summarization task, while CNN/DM and Guardian data set still outperforms the baseline we reported in Table 1. This model struggles with generating meaningful sequences. As reported in Table 2, we can see that this is the case for all the data sets we have used. It seem that solely training on language modeling is not very effective, though it might be a helpful addition to the other model. A possible reason is that the N-gram is too large for some sentences, sometimes needing to calculate the probabilities for 10 or more subsequent words.

Overall, the results show that it is possible using pre-training techniques to produce summaries that capture the core essence of a document. In addition, producing grammatically correct and fluent summaries is also achievable even though this is not the case for all summaries. However, pre-training on the learning objective of language model does not seem to generate good results. Also, the results show that there are still a large occurrence of limitations for the generated summaries.

7 CONCLUSION

In this paper, we have explored applying transfer learning through to the task of text summarization. We made an attempt to research whether the performance level of a small target domain can be improved using pre-training techniques. We first pre-trained three

data sets on two different learning objectives, namely summarization and language model. This resulted in six different models that were afterwards fine-tuned on a smaller target data set consisting of blog articles.

The results show that pre-training it with the summarization task vastly outperforms the baseline performance we defined in Section 4.3. Upon further manual inspection, it can be seen that these models are able to generate summaries that capture the core essence of an article. However, most of the generated summaries are lacking in readability due to their inconsistency and incorrectness in grammar. In addition, some generated summaries suffer from OOV words, resulting in [UNK] tokens. This may be caused by using the smaller vocabulary, a methodology we adopted from the work of other authors.

7.0.1 Future work. Further research should be undertaken to investigate how to select a data set to use as source task for transfer learning. Methods to assess the relatedness between source-and target task could be investigated, as the success of transfer learning often depends on this. For example, a clustering technique can be used to compare the similarity of two data sets.

In this research we used the fine-tuning approach without freezing layers. We believe that using other fine-tuning strategies might improve the performance of this model. For example, freezing only a few layers instead of retraining all the parameters as often done in image recognition, might have a strong difference in the results. Also, further research is needed to assess how adjusting the learning rate during fine-tuning affect the performance.

In addition, we attempted to try pre-training on the learning objective of language model, similarly to some other work that are state-of-the-art [7, 16]. This did not result in a good performing model, which we believe may be caused by some incorrectness in the implementation.

8 ACKNOWLEDGEMENTS

Firstly, I want to thank my UvA internal supervisor dr. Pengjie Ren for the precious feedback and guidance during the thesis period. I am especially thankful for the time and commitment put in supervising the students with valuable weekly meetings. Also I am grateful for the flexibility and freedom I was granted in the research, as he willingly adapted to my wishes. Secondly, I appreciate the support I had received from Accenture through my supervisor Ruben Gort, my buddy Patrick Bossen, the data talents program team for interns; Matthijs, Richard, Virgile, Randy, Umo, Jesper, Ron, Chiara, Cleo and other Accenture colleagues; Shujun, Shuhui, Gary, Moegi, Nahm, and Nicky. Lastly, a special thanks to dhr. Nikos Voskarides for the willingness and enthusiasm of being my second reader.

REFERENCES

[1] Stergos Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. 2005. Summarization from medical documents: a survey. *Artificial intelligence in medicine* 33, 2 (2005), 157–177.

[2] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268* (2017).

[3] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[4] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252* (2016).

[5] Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 93–98.

[6] Dipanjan Das and André FT Martins. 2007. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU* 4, 192-195 (2007), 57.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[9] Atefeh Farzindar and Guy Lapalme. 2004. Letsum, an automatic legal text summarizing system. *Legal knowledge and information systems, JURIX* (2004), 11–18.

[10] Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *16th International joint conference on artificial intelligence (IJCAI 99)*, Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 668–673.

[11] Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47, 1 (2017), 1–66.

[12] René Arnulfo García-Hernández, Romyna Montiel, Yulia Ledeneva, Eréndira Rendón, Alexander Gelbukh, and Rafael Cruz. 2008. Text summarization by sentence extraction using unsupervised learning. In *Mexican International Conference on Artificial Intelligence*. Springer, 133–143.

[13] Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence* 2, 3 (2010), 258–268.

[14] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*. 1693–1701.

[15] Eduard Hovy, Chin-Yew Lin, et al. 1999. Automated text summarization in SUMMARIST. *Advances in automatic text summarization* 14 (1999).

[16] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[17] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[18] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).

[19] Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*. Association for Computational Linguistics, 45–51.

[20] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111* (2016).

[21] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[22] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* (2016).

[23] SJ Pan and Q Yang. 2010. A survey on transfer learning. IEEE Transaction on Knowledge Discovery and Data Engineering, 22 (10).

[24] Foster Provost and Tom Fawcett. 2013. Data science and its relationship to big data and data-driven decision making. *Big data* 1, 1 (2013), 51–59.

[25] Avinesh P.V.S., Maxime Peyrard, and Christian M. Meyer. 2018. Live Blog Corpus for Summarization. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*. 3197–3203. http://www.lrec-conf.org/proceedings/lrec2018/pdf/317.pdf

[26] Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Liqiang Nie, Jun Ma, and Maarten De Rijke. 2018. Sentence relations for extractive summarization with deep neural networks. *ACM Transactions on Information Systems (TOIS)* 36, 4 (2018), 39.

[27] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, Vol. 898. 1–4.

[28] Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*. Springer, 3–21.

[29] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* (2017).

[30] Fei Song and W Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*. ACM, 316–321.

[31] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association.*

[32] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 242–264.

[33] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811* (2016).

[34] Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *Proc. of aaai email-2008 workshop, chicago, usa.*

[35] Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 1225–1237.

[36] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 9.

[37] Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 985–992.

[38] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[39] Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and Ming Zhou. 2019. Pretraining-Based Natural Language Generation for Text Summarization. *arXiv preprint arXiv:1902.09243* (2019).