Automatic Text Summarization through global semantics

Andi Zhang*, Andrew Monnin†, Anh Vu‡ and Bhumika Bhatt§
College of Computing
Georgia Institute of Technology
*azhang6@gatech.edu, †amonnin3@gatech.edu, ‡avu41@gatech.edu, §bbhatt8@gatech.edu

***Abstract*—In this paper we examine a way to improve the performance of current state-of-the-art abstractive summarization models. We implement and evaluate the effectiveness of document-related modulation (DRM), a key component of the topic assistant module architecture introduced in "Friendly topic assistant for transformer based abstractive summarization" (Wang et al., 2020). This architecture improves upon the global semantics of current state-of-the-art abstractive summarization models by first identifying topics in each document using a topic model. The topic proportion vector generated from a topic model is then fed into the DRM layer present in the base, state-of-the-art model's decoder. This is done to improve the global semantics of the generated summary through feature biasing. Due to the importance of the topic model to this approach, we experiment with two alternative topic models, the Latent Dirichlet Allocation (Blei et al., 2003) and the Weibull Hybrid Autoencoding Inference (Zhang et al., 2020) to see how each affects the performance of the summarization task.**

I. Introduction

Automatic text summarization is understanding a document and then generating a text. It is of two types - extractive summarization and abstractive summarization. Extractive summarization approaches involve identifying representative sentences and concatenating them into a summary. The more challenging out of the two, the abstractive summarization generates a summary by choosing new words and phrases, using phrases from the source text and rephrasing them. Transformer-based models have shown state-of-the-art performance in abstractive text summarization. These models are first pre-trained on a large corpus and then fine-tuned for summarization tasks. While these models are good at understanding relationships between local tokens, they lack understanding of the global semantics. Thus, these models tend to favor encoding short-range dependencies (Zang et al., 2020)

A solution to provide the model with global semantics is using probabilistic topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) or Poisson Factor Analysis (PFA) (Zhou et al., 2012).

Wang et al. have implemented and shown better performance of a topic model based, topic assistant (TA) for transformer based abstractive summarization.

TA is user friendly plug-and-play model which is compatible with many transformer-based summarization models. Here the user can simply fine tune the transformer and TA using any pretrained model. Additionally, TA introduces only a few extra parameters compared to the base model. The original paper have employed pretrained BertSUM (Liu et al., 2019) transformer-based model along with PFA. In this work, we have tried to implement one of the three components of TA model, called Document-related modulation combined with the base model obtained from (Liu and Lapata, 2019).

We have used CNN / DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016) in our experiment. The dataset contains over 300,000 news articles which are written by journalists at CNN and DailyMail. The articles from CNN dates from April 2007 to April 2015, while DailyMail articles date from June 2020 to April 2015. The original version (version1.0.0) was developed for machine reading and comprehension as well as for abstractive question answering. The later versions 2.0.0 and 3.0.0 are used for abstractive and extractive summarization. Version 3.0.0 is a non-anonymized version while the previous two versions were anonymized where the named entities were replaced with unique identifier labels. In our work we have employed the same version (version 3.0.0) as employed by Liu and Lapata, 2019 and by Wang et al,2020 in their studies.

The dataset is split into train, validation and test sets. The dataset contains three fields – id, article -which contains body of the article, and highlights - which contains highlight of the news article written by the author of the article. This highlight is what we used to judge the quality of the summaries generated by our model. The original article was fed into the encoder as the raw document, both during training and testing/validation. While the highlight (summary) was fed into the decoder during training and used to judge the model output with during testing/validation.

II. Approach

A. Topic Assistant and our approach

At its core, the Topic Assistant (TA) model consists of a transformer-based model fine-tuned for abstractive text summarisation task. In both our implementation and the "The Friendly Topic Assistant Model", we used the architecture of BertSum (Yang et al., 2019) as our base fine-tuned model. We then improve the performance of this model by incorporating important global topics of the document through several plug-and-play modules. Specifically, in the original implementation of TA, there are three main modules. They are the semantic-informed attention (SIA), which is implemented in the self-attention module; the topic embedding with masked attention

(TEMA), which is implemented after the token and positional embedding in the decoder; and finally the document-related modulation (DRM), which is implemented after the cross-attention module in the decoder.

Based on the paper's ablation study, TEMA and DRM are able to achieve better improvements in model's summarization performance compared to SIA (Wang et al., 2020). This is because SIA only improves the the local relation in the Transformer attention module, while the DRM and TEMA are better at introducing global semantics into the Transformer-based models, thus better at improving the summary results. Additionally, in most cases, DRM are able to attain similar or better improvements in model performance than TEMA, as shown in the Table II. Because DRM are able to perform relatively best out of all three modules, we focus on building the DRM module to improve the model performance. A key input of this module is θ, which is the topic distribution for each document. We examine both LDA and WHAI as a method to create this topic proportion vector θ.

Model	R1	R2	RL
BertSUM	42.13	19.60	39.18
BertSUM + SIA	42.48	19.99	39.37
BertSUM + TEMA	42.77	20.12	39.46
BertSUM + DRM	42.66	20.33	39.56

TABLE I
ABLATION STUDIES, BASED ON BERTSUM AND CNN/DM DATASET (WANG ET AL., 2020)

B. BertSUM

Bidirectional Encoder Representations from Transformers (BERT) (Delvin et al., 2018) is a Transformer based model that has achieved a lot of success in performing different NLP tasks. Prior to BERT, language models generate token by looking at the text sequence from left-to-right or a combination of left-to-right and right-to-left. This one directional approach works by predicting the next words, append it to a sequence and repeat the process. BERT is very powerful compared to prior language models since it is bidirectionally trained, thus having a much better sense of the flow and language context compared to uni-direction language models. Additionally, BERT uses the Masked Language Modeling (MLM) technique, which randomly masks context words in the sentence and force the model to predict it using the words in both directions. This methods of masking enhance the BERT ability to understand the underlying context of the sentence. Furthermore, it reduces the amount of data needed for training. As each document can be used to train from both directions.

By building on the success of Bert, Yang et al. introduced BertSUM as a Transformer-based summarization model for both abstractive and extractive summarization tasks (2019). BertSUM inherits the same encoder-decoder architecture of a Transformer-based model. Given a data pair x, y, where the the document x has N_1 tokens, and the summary y has N_2 tokens, the BertSUM model generate summary by maximizing the likelihood equation below, with x_k and y_k denote the $k-th$ token in the document and summary:

$$\prod_{j=1}^{N_2} p(y_j|\{x_i\}_{i=1}^{N_1}, y_{i<j}) \tag{1}$$

The encoder of BertSUM is a twelve-layer pre-trained BERT with bidirectional self-attention module and feed-forward network, with addition and normalization after each. The decoder of BertSum is six Transformer decoder layer, with each layer containing a self-attention, cross-attention module, and feed-forward network, with addition and normalization after each module.

C. Document-related modulation

Feature biasing is an effective method to implement certain conditions into the model. The authors introduce the document-related modulation (DRM) module as a way to implement feature bias in the transformer's decoder. The DRM uses the topic proportion vector θ from the topic model to introduce topic bias to the output of the cross-attention module using the equation below:

$$z = \theta^T W_b \tag{2}$$

W_b is the parameter matrix in the DRM, with the dimension of the number of topics from the topic model by the dimension of the self-attention module. The bias vector z is then added to the output of the cross-attention module before the addition and normalization operation. DRM module introduces additional parameters, which increase the time of fine-tuning the model. Regardless, the number of additional parameters from DRM is very small compared to the original model, and thus doesn't sacrifice too much learning and test speed. Specifically, the DRM only introduce an additional of 1.7 percent of new parameters to the BertSUM base model (Wang et al., 2020), while SIA introduces 3.95 percent of new parameters.

D. Topic models: Latent Dirichlet Allocation and Weibull hybrid autoencoding inference

Topic models learn topics from a corpus in an unsupervised way. In our experiments, we use two different topic models to incorporate with the DRM, i.e., the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and the Weibull hybrid autoencoding inference (WHAI) (Zhang et al., 2020). LDA is a generative probabilistic topic model. It is based on the objective that every document is a probability distribution over topics and every topic is a distribution over words. Thus, one can find hidden topics from the collection of words that frequently co-occur. LDA loss function is the log likelihood with respect to the model parameters (Blei et al., 2003).

On the other hand, WHAI is a hybrid Bayesian inference for deep topic modeling that integrates both stochastic-gradient Markov Chain Monte Carlo (MCMC) (Welling Teh, 2011; Ma et al., 2015; Cong et al., 2017a) and a multilayer Weibull distribution based VAE. WHAI is related to a VAE in having both a decoder and encoder, but differs from a usual VAE

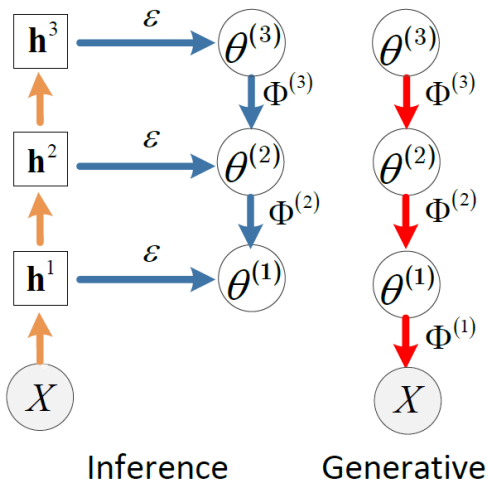

Fig. 1. WHAI architecture (Zhang et al., 2020)

in the following ways: 1) deep latent Dirichlet allocation (DLDA), a probabilistic deep topic model equipped with a gamma belief network, acts as the generative model; 2) inspired by the upward-downward Gibbs sampler of DLDA, the inference network of WHAI uses a upward-downward structure, to combine a non-probabilistic bottom-up deep NN and a probabilistic top-down deep generative model, with the n^{th} hidden layer of the generative model linked to both the $(n+1)^{th}$ hidden layer of itself and the n^{th} hidden layer of the deep NN; 3) a hybrid of stochastic-gradient MCMC and autoencoding variational inference is employed to infer both the posterior distribution of the global parameters, represented as collected posterior MCMC samples, and a VAE that approximates the posterior distribution of the local parameters given the data and a posterior sample (rather than a point estimate) of the global parameters; 4) the Weibull distributions is used in the inference network to approximate gamma distributed conditional posteriors, exploiting the fact that the Weibull and gamma distributions have similar probability density functions (PDFs), the Kullback-Leibler (KL) divergence from the Weibull to gamma distributions is analytic, and a Weibull random variable can be efficiently reparameterized with a uniform noise.

The use of the Weibull distribution, which resembles the gamma distribution and has a simple reparameterization, makes one part of the evidence lower bound (ELBO) analytic, and makes it efficient to compute the gradient of the non-analytic part of the ELBO with respect to the parameters of the inference network. Moving beyond deep models and inference procedures based on Gaussian latent variables, WHAI provides posterior samples for both the global parameters of the generative model and these of the inference network, yields highly interpretable multilayer latent document representation, is scalable to a big training corpus due to the use of a stochastic-gradient MCMC, and is fast in out-of-sample prediction due to the use of an inference network. The loss function of the variational autoencoder is the negative log-likelihood with a regularizer.

E. LDA Implementation

Following steps were followed to build an LDA model:

1) Text pre-processing: This involved tokenization, lemmatization and stopwords removal. In this step, the text was first broken into constituent words and then part-of-speech (POS) tagging was done.POS tagging helped us in extracting words that are noun, adjective, proper noun and verbs and we ignored other words that are not be useful for our current task. These filtered words are then lemmatized. This helps in converting plural words such as 'books' into singular form 'book'. We also removed stopwords. These stopwords such as 'and' and 'the' do not provide any differentiating meaning to the text. Thus, stop words removed. We have used sPacy, which is a python library for NLP tasks.
2) Building LDA model: We first built a vocabulary from the words derived from the pre-processing step. This vocabulary helps in building a document-term-matrix where each row is a document and each column corresponds to a word. Here, each cell represents the number of times a particular word occurs in the corresponding document. This is bag-of-words(BOW) model and is the input to our LDA model. Additionally, the user has to provide the number of topics she/he expects in the corpus and then LDA builds a model with those many topics. A low number of topics can produce broad topics and a high number of topics can produce topics that have repeated words. Considering our dataset contains news articles we expected the number of topics to be around 100s. In a preliminary run, using 256 number of topics (as suggested by (Wang et al., 2020), our topics were coming with many repeated words. As there was only a small difference in Rouge scores between model with 128 and 256 topics (Wang et al., 2020),we selected 128 topics to build our LDA model. LDA model was implemented using Gensim library.

In order to solve the question of how to integrate output from LDA model with the BertSum model, following strategy was employed. As mentioned above, we needed topic distribution for each document (θ) for integration with BertSum model.

The above mentioned steps of text pre-processing and building LDA model were used to first produce processed data. This was used to build an LDA model on the entire training dataset. This topic model was then queried to extract theta. The dataset was then augmented with the theta corresponding to each document. We then used this theta-augmented dataset to fine-tune our base BertSum model.

While building LDA model we encountered the problem with the slow text pre-processing speed associated with sPacy library and the long time taken in building LDA model. We addressed this issue in the following steps:

1) We disabled multiple features of sPacy library that we do not need, only keeping the essential ones.

2) We used pipeline functionality of sPacy. This sets up the pipeline of the sPacy components that are needed for text pre-processing and helps in faster processing of the dataset.
3) We used LDAMulticore from Gensim library instead of plain LDA,as LDAMulticore parallelizes and speeds up LDA model training.
4) We used a library called joblib for parallelization and speeding up the LDA model training.

All these steps drastically reduced the LDA model building time from expected 80 hours to 3 hours on a local machine with 4 cores.

F. WHAI Implementation

We leverage the source code of the WHAI paper [1] and adapted it to train the model on the CNN / DailyMail dataset. The paper did not specify any preprocessing for the model. So we use the same tokenizer with the pre-trained BertSUM provided by Yang Liu without any modification. It turned out the loss would not decrease stably with this setting. The loss would even oscillate with a little higher learning rate (0.0005), instead of using the default learning rate of 0.0001. Some higher learning rate would even lead to infinity values for the loss. We then used the same tokenizer with our base model, but removed the stop words and short tokens (less than two characters, with that we removed all the punctuation). In addition, we lemmatized all the documents as well. With all those changes, the training of WHAI on CNN / DailyMail is much faster and smoother. We implemented the WHAI with 64 topics and 128 topics and obtained the document-topic distribution θ. WHAI was implemented using the Theano framework.

G. BertSUM and DRM Implementation

For our base Transformer mode, we leverage the source code [2] of pre-trained BertSUM provided by Yang Liu. The BertSUM source code used PyTorch for its implementation. The dataset is provided in the same repository, but can also be accessed from here [3].

First, we implement the topic model to get the topic proportion vector θ for our DRM as in equation (2). Our approach differs in the original authors' in that instead of using PFA as our the topic model, we implement LDA and WHAI instead.

Finally, we implement the DRM in the decoder of the base model. For both WHAI and LDA based model, the weight W_b in equation (2) is initialized randomly between 0 and 1. These are new parameters that we need to train from scratch in our training process. The weight is multiplied with the θ from the topic model to create the feature bias. This feature bias is then added to the output of the cross-attention module in all 6 decoders layers of the Transformer. Our model structure is shown in Fig 2. By incorporating the DRM, the number of parameters for BertSUM increased to approximately 190 million.

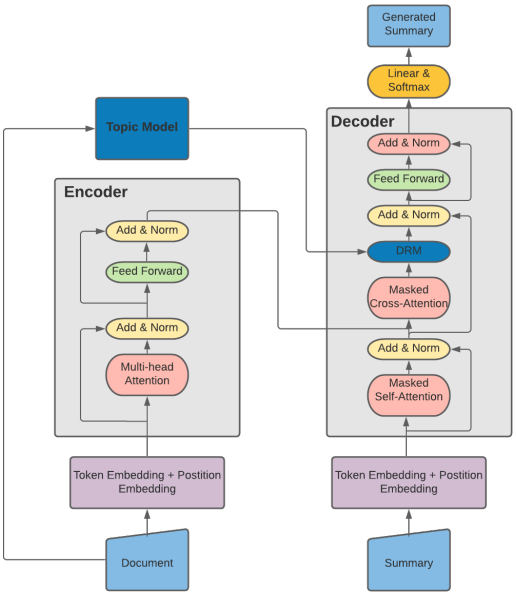

Fig. 2. BertSum with TA. We use LDA and WHAI as the topic model

By implementing the above mentioned step, we have produced two models:

1) WHAI providing the θ for DRM in BertSUM (WHAI + BertSUM)
2) LDA providing the θ for DRM in BertSUM (LDA + BertSUM)

III. Experiments, Results, and Analysis

A. Experiments setup

We first train the topic model LDA and WHAI to obtain the topic proportion vector θ. When running WHAI using Adam optimizer, we experiment with hyperparameters such as different values of topic, including 64 and 128, different learning rates to obtain the different value of vector θ. Due to the limited computing power, we only able to use 5 percent of the training data for WHAI. However, for LDA, we use the whole training dataset. For LDA, we use 128 topics for our implementation. After training the topic model, we plug the obtained θ into the DRM module.

Because the weight in DRM is newly intialized, these parameters are trained from scratch during training time. We then fine-tune the BertSum model along with the DRM and compare the performance with the base pretrained BertSum. During our fine tuning process, we used the default parameters provided in the source code [4], and experimented with different number of steps to see how it affects our model performance.

For training and testing data, we use the data from the CNN / DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016) for training the WHAI and LDA, finetuning

[1]https://github.com/BoChenGroup/WHAI
[2]https://github.com/nlpyang/PreSumm
[3]https://drive.google.com/file/d/1DN7ClZCCXsk2KegmC6t4ClBwtAf5galI/view [4]https://github.com/nlpyang/PreSumm

the BertSum, and testing the BertSum with DRM incorporated. We train the WHAI and LDA, and then fine-tune the Transformers-based model using 10000 documents (less than 5 percent of the total training dataset). We then test our results on 4000 documents to assess performance between different topic model configurations. Due to limited computing power, we train the BertSUM with DRM implementation with 11000 steps.

To evaluate the quality on summarization, we use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score, which is also used in the original paper. Specifically, we compare the ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) score, which calculates the overlap of unigram, bigrams, and the longest common subsequence between the generated and reference summaries, respectively, to assess fluency.

B. Results and analysis

Below are the ROUGE scores we obtained from our top performing models.

Model	R1	R2	RL
Base	36.05	15.10	0.2725
Base + DRM (WHAI with 128 topics)	36.78	15.84	28.46
Base + DRM (WHAI with 64 topics)	36.64	15.84	28.14
Base + DRM (LDA)	36.68	16.00	28.39

TABLE II
EXPERIMENT RESULTS WITH DIFFERENT TOPIC MODEL AND NUMBER OF TOPICS

We successfully implemented the DRM in a manner which improved upon the base ROUGE scores across some or all of the rouge metrics (Table 2), dependent on the particular DRM implementation. This shows that the DRM is able to incorporate feature topic bias when generating the summary. Furthermore, anecdotally we are able to find examples where our implementation clearly gave a better summary then the original model, as shown below. However, there were other cases where the original seemed to have the better summary and in the majority of cases there was no material difference. This makes sense as the topic model only improves performance when there is value added from adding a global context. If the base model already included these key topics then there would be little change. And conversely, if the most important part of the document is centered on one section then adding the global context can degrade performance. Another important difference to note is that LDA was trained on the full CNN Daily mail dataset while WHAI was only trained on 5% of the same dataset due to compute and time constraints.

Below is an example table A where each of the topic models was able to pickup on key information that the base model missed. Namely that a key player was injured but never-the-less was able to play the full game. Furthermore the base models summary is not as coherent as each of the topic models. We see that both WHAI 128 Topics and WHAI 64 topics were very similar to each other in both content and structure, with slight differences in some of the auxiliary information it picks up. Meanwhile LDA was able to pickup on an entirely different topic, namely the teams score from throughout the rest of the season. All of this additional information is scattered throughout the entirety of the document, so it makes sense that the DRM, with its focus on global info (See appendix for original source document).

BertSum *daryl janmaat has admitted newcastle got exactly what they deserved.newcastle lost 1-0 to sunderland in the tyne-wear derby on sunday.newcastle were not at the races as a sunderland side motivated by new head coach dick advocaat*

BertSum+WHAI-128 *newcastle united lost 1-0 to sunderland at the stadium of light on sunday.the holland full-back played the full 90 minutes despite a suspected tear in his calf muscle during the warm-up.janmaat has admitted that newcastle got exactly what they deserved from their defeat*

BertSum+WHAI-64 *newcastle lost 1-0 to sunderland in the premier league on sunday.daryl janmaat played the full 90 minutes despite a suspected tear in his calf muscle during the warm-up.the dutch full-back has admitted that newcastle got exactly what they deserved*

BertSum+LDA *sunderland beat a lacklustre newcastle 1-0 in tyne-wear derby.janmaat played the full 90 minutes despite suspected calf injury.the holland full-back admitted performance was n't up to standard.black cats have not taken 17 of last 21 point available in derby games*

Table A - Summary obtained from different models

IV. LIMITATIONS AND FUTURE WORKS

Due to the complexity and size of the model, the biggest problem that we faced was the computation time.

While our results show that the BertSUM model with DRM is somewhat able to generate summary with global topic, there were several limitations in our implementation and experiments, which will allow future works to achieve even better results in their implementation. First, with regards to the training process of the topic models, it is possible to do additional hyper-parameters tuning for WHAI. Specifically, we can experiment with a wider range of number of topics besides on 64 and 128 to see how it affect the summarization, as well as try adding more layers into WHAI to see how it might improve the summarization results.

Additionally, we only able to use less than 5 percent of the data of the data due to limited access to the computing power. Specifically, AWS took longer than expected to give us access to GPU instances, and as a result we have to train and test the model with a combination of personal computer and Google Colab pro, which already took more than 10 hours in total. In the future, we can train with more data for a longer time to see how effective the DRM module is in improving the base model performance. Additionally, we can experiment with different dataset such as XSum, to see how well the model generalize.

We also notice that pure ROUGUE score is not a good metric to measure the performance of abstractive summarization model. At its core, pure ROUGE score measures how many words in the gold summaries exist in the model generated summaries. For future work, we can try using different metrics

to compare the performance of our model. Two candidates for this is the BLEU score, which measures how many words in the model generated summaries exist in the gold summaries, or F1 score, a combination of the BLEU and ROUGE.

The impact of the success has we found here is quite large. There are numerous real world applications for high performing abstractive sumarisation, from the task we trained on, summarising new articles, to giving the highlights of legal documents. While there is still work to do until automatic abstract summarisation is on par with a human, even its current-state it still has the potential to be quite powerful and wide-reaching.

V. Work Division

Please find the table with more information on the delegation of work among team members in Table **??**. Our code repo can be found here [5]. Please reach out amonnin3@gatech.edu if you are unable to access this repo

References

[1] H. Zhang, B. Chen, D. Guo, and M. Zhou, "Whai: Weibull hybrid Autoencoding inference for deep topic modeling," arXiv.org, 25-Apr-2020. [Online]. Available: https://arxiv.org/abs/1803.01328.

[2] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," PMLR, 21-Nov-2020. [Online]. Available: https://proceedings.mlr.press/v119/zhang20ae.html. https://www.overleaf.com/project/61a538e13f5e355c21da0bf9

[3] M. Zhou, L. Hannah, D. Dunson, and L. Carin, "Beta-negative binomial process and Poisson factor analysis," PMLR, 21-Mar-2012. [Online]. Available: https://proceedings.mlr.press/v22/zhou12c.html.

[4] Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022

[5] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," arXiv.org, 05-Sep-2019. [Online]. Available: https://arxiv.org/abs/1908.08345.

[6] Z. Wang, Z. Duan, H. Zhang, C. Wang, L. Tian, B. Chen, and M. Zhou, "Friendly topic assistant for Transformer based abstractive summarization," ACL Anthology. [Online]. Available: https://aclanthology.org/2020.emnlp-main.35/.

[7] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Neural Information Processing Systems.

[8] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, C¸ aglar Gulc¸ehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Computational Natural Language Learning.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional Transformers for language understanding," ACL Anthology. [Online]. Available: https://aclanthology.org/N19-1423/.

[10] Y. Liu, "Fine-tune bert for Extractive Summarization," arXiv.org, 05-Sep-2019. [Online]. Available: https://arxiv.org/abs/1903.10318v2.

[11] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In ICML,pp. 681–688, 2011.

[12] Y. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient MCMC. In NIPS, pp. 2899–2907, 2015.

[13] Mingyuan Zhou, Yulai Cong, and Bo Chen. The Poisson gamma belief network. In NIPS, pp.3043–3051, 2015.

Appendix

Orginal Source Document Example

efender daryl jan has admitted newcastle got exactly what they deserved from their derby trip to sunderland - nothing . holland full ack played the full 90 minutes of sunday barclay premier league defeat at the stadium of light despite suffering a suspected tear in his calf muscle during the warm p , but was unable to prevent the black cats running out winners as the mag pies slipped to a record fifth successive defeat in the fixture . newcastle created little of note on another poor day for john carver men , and jan was pulling no punches as he assessed the fallout . [SEP] [CLS] daryl jan ma at (left) has admitted that newcastle united ' s performance in their tynewear derby against sunderland was n good enough and they got exactly what they deserved - nothing [SEP] [CLS] the holland full back played the full 90 minutes despite a suspected calf muscle tear [SEP] [CLS] jan tries to get across and stop a cross from sunderland ' steven fletcher at the stadium of light [SEP] [CLS] he told nu v : it was not good enough . [SEP] [CLS] the first half was really poor , we were too negative , so it was not good enough . [SEP] [CLS] the second half was much better , but it still was n good enough . ' [SEP] [CLS] i would n say we did n give everything . [SEP] [CLS] what i can say is we were not good enough and everybody saw that sunderland were better , especially in the first half . [SEP] [CLS] ' if you are not good enough , then most of the time , you lose . [SEP] [CLS] newcastle were simply not at the races as a sunderland side motivated by new head coach dick ad vo ca at dominated from the off , although they looked like making it to half time unharmed before je smashed home a stop time volley which proved to be the tangible difference between the teams . [SEP] [CLS] jan a t said : ' a goal in the last minute of the first half , it always hard to con cede it because you go into the dressing room with a bad feeling . [SEP] [CLS] if you go in at 0 , you can be positive in the dressing room , but if you con cede a goal in the 47th minute , then it ' s hard . [SEP] [CLS] jerma n def oe scored the winning goal - a spectacular volley - on the stroke of half time [SEP] [CLS] def

[5] https://github.gatech.edu/amonnin3/dl-textsummary