Advanced Natural Language Processing Project REPORT

Mylavarapu Maheswara Rao
IIIT-Hyderabad, India
`maheswara@students.iiit.ac.in`

Anubhav Sharma
IIIT-Hyderabad, India
`anubhav.sharma@iiit.ac.in`

[

2]

Abstract

The work done and analysed in this report is derived from [3]. The task of abstractive summarising is one of the most intriguing tasks in Natural Language Processing(NLP) and a necessity , for highlighting the key points in a text and making information more accessible and comprehensive . Over that , query-based summarization highlights those points that are relevant in that particular query , thus enhancing the quality of the summary generated with respect to the user . Here we will use the encoder decoder model , quite popular in the downstream tasks of machine translation, abstractive summarization, dialog systems, etc. but here with three different approaches we make an attempt to reduce the model's tendency to generate repetitive sentences.Following are the approaches :-
(i) a query attention model (in addition to document attention model) which learns to focus on different portions of the query at different time steps (instead of using a static representation for the query) and (ii) a new diversity based attention model which aims to alleviate the problem of repeating phrases in the summary.
The dataset created in our approach is inspired by Preksha Nema's Approach [3] , the query-based summarization dataset building on debatepedia . It involved extraction of text from their API and forming of triples .

1 Introduction

The approach of encoder decoder models [1] have outperformed many other models when it comes to downstream tasks related to Natural Language Generation(NLG) . There was some prior work in the field of extractive summarisation where the sentences containing the most salient features were selected and displayed as a summary , which was not the case in abstractive summarisation.So first step involved was to create a dataset . We used debatepedia to create this dataset and it contained triplets of the form (query, document, summary) and each summary is abstractive. This dataset contains triplets of the form (query, document, summary). Further, each summary is abstractive and not extractive in the sense that the summary does not necessarily comprise a sentence which is simply copied from the original document.

When it comes to vanilla encoder-decoder models , as discussed in the paper [2] , the summaries generated contains a lot of repetitive words , thus further degrading the quality of the summary . A typical encode-attend-decode model first computes a vectorial representation for the document and the query and then produces a contextual summary one word at a time.So to ensure that our decoded words are not the same as there previous words , we check that the context vector is orthogonal to the immediate previous context vector (this leaves a possibility to attend a word in later part of the summary generation) check if the hidden state's value generated after passing the context vector through LSTM is orthogonal to immediate previous hidden state's value . This hidden state value is fed to the decoder to generate the word .

Following are the tasks that we have performed in this project :- (i) Using the technique to create the dataset as mentioned in [3], we created the dataset for query based abstractive summarization . (ii) Propose the solution approaches for NLG and outperforming the baselines by almost 7-8% & outperforming the scores achieved by [3] by almost 20%.(iii) qualitatively analysed the results and showed that our model indeed produces outputs with fewer repetitions(similar to [3].

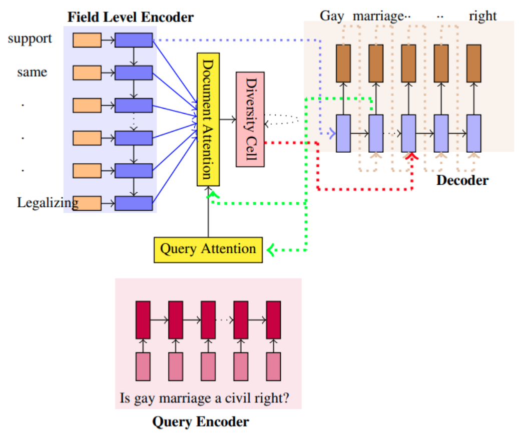
Let us look at an example summary that our model predicted : *Query* - `<s> homophobia : will the children of gay parents be able to deal with societal homophobia ? <eos>` *Output* - `homosexuals are taught a tolerance toward dysfunctionality` .

2 Literature Review

For understanding and understanding the complete idea behind the extractive summarisation process , we went through the survey by Nenkova [4].One recurring problem in encoder-decoder models for NLG is that they often repeat the same phrase/word multiple times in the summary (at the cost of both coherency and fluency).In his approach Sankaran [5] studies this problem in the context of MT and propose a temporal attention model which enforces the attention weights for successive time steps to be different from each other. Thus giving the intuition for attention based approach in the architecture of [3].
Chen's proposal[2] of a distraction based attention model which maintains a history of attention vectors and context vectors. It then subtracts this history from the current attention and context vector. This didn't work well on the dataset for [3] , the reason being agressive discounting on history terms & this finally rose the idea for a LSTM , as it had a better mechanism for dealing with history (with forgot gate).

3 Dataset

The dataset was retrieved in a similar fashion as done in Preksha Nema's paper [3]. The dataset was curated specially for query based summarisation and was extracted from debatepedia.Debatepedia is an encyclopedia of pro and con arguments and quotes on critical debate topics.12695 triples of form query, document, summary (which were present at the debatepedia server) were extracted from debatepedia .
They used 10 fold cross validation for all their experiments. Each fold used 80 % of the documents for training, 10% for validation and 10% for testing.



3.1 Model

Let q be a query,such that, **q**=$q_1, q_2, \ldots, q_k$ containing k words and let document **d**=$d_1, d_2, \ldots, d_m$.Let contextual summary be y such that **y** = $y_1, y_2, \ldots, y_l$ containing l words.This can be modeled as the problem of finding y^* that maximizes the probability $p(y \mid$ **q,d**) which can be further decomposed as:

$$y^x = \arg\max_y \prod_{t=1}^{m} P(y_t|y_1, y_2.., y_{t-1}, \mathbf{q}, \mathbf{d}) \quad (1)$$

Encoder for the query: They used a recurrent neural network with Gated Recurrent Units (GRU) for encoding the query.It reads the query from left to right as $q_1, q_2, \ldots, q_k$. In our approach we used a multilayer Bidirection Gated Recurrent Units (GRU) for encoding the query.

$$h_i^q = GRU_q(h_{i-1}^q, e(q_i)) \quad (2)$$

where e(q_i) ϵR^d is the d-dimensional embedding of the query word q_i
. **Encoder for the document:** This is similar to the query encoder and reads the document **d** =$d_1, d_2, \ldots, d_k$ from left to right. Here again we used the multilayer BiDirectional GRU based architecture .

$$h_i^d = GRU_d(h_{i-1}^d, e(d_i)) \quad (3)$$

where e(d_i) ϵR^d is the d-dimensional embedding of the query word d_i
. **Attention mechanism for the query :** At each

time step, the decoder produces an output word by focusing on different portions of the query (document) with the help of a query (document) attention model.

$$a^q_{t,i} = v^T_q tanh(W_q s_t + U_q h^q_i) \tag{4}$$

$$\alpha^q_{t,i} = \frac{exp(a^q_{t,i})}{\sum_{j=1}^{k} exp(a^q_{t,j})} \tag{5}$$

$$q_t = \sum_{i=1}^{k} \alpha^q_{t,i}.h^q_i \tag{6}$$

where s_t is the current state of the decoder at time step t (we will see an exact formula for this soon). $W_q \epsilon R^{l_2 l_1}, U_q \epsilon R^{l_2 l_2}, v_q \epsilon R^{l_2} l_1$ is the size of the decoder's hidden state, l_2 is both the size of h^q_i and also the size of the final query representation at time step .

Attention mechanism for the document : The document attention model works in a way in which it assigns weights to each word in the document & query using the following equations.

$$a^d_{t,i} = v^T_d tanh(W_d s_t + U_d h^d_i + Z q_t) \tag{7}$$

$$\alpha^d_{t,i} = \frac{exp(a^d_{t,i})}{\sum_{j=1}^{k} exp(a^d_{t,j})}$$

$$d_t = \sum_{i=1}^{k} \alpha^d_{t,i}.h^d_i \tag{8}$$

where st is the current state of the decoder at time step t (we will see an exact formula for this soon). $W_d \epsilon R^{l_4 * l_1}, U_d \epsilon R^{l_4 * l_4}, Z \epsilon R^{l_4 l_2}, v_d \epsilon R^{l_2}.l_2, l_4$ is the size of $h^i_d$and also the size of the final document representation d_t which is passed to the decoder at time step t as As we can note now that d_t encodes the relevant information from the document as well as the query.d_t of equation(8) is context vector for decoder.

Decoder: The hidden state of the decoder s_t at each time t is again computed using a GRU as follows: This is exactly the same quantity defined in Equation (1) that we wanted to model , i.e Pr(y$_t$ | $(x_1, x_2, \ldots, x_n,$**q,d**$)$).

$$y_t, s_t = GRU_{dec}(s_{t-1}, [e(y_{t-1}), d_{t-1}]) \tag{9}$$

Where s_t is state of decoder and $[e(y_{t1}), d_{t1}]$ means a concatenation of the vectors $e(y_{t1}), d_{t1}$. This was the basic encoder decoder model. This model suffers from the problem of repeating the same phrase/word in the output so now we will move forward to the novel diversity based attention model for query based summarisation as proposed by [3].

3.2 Diversity based attention models

If the decoder produces the same phrase/word multiple times then it is possible that the context vectors being fed to the decoder at consecutive time steps are very similar . There were 4 approaches that they used :- D_1, SD_1, SD_2 .

D1: D1: In this model, after computing dt as described in Equation (8), we make it orthogonal to the context vector at time t 1:

$$d'_t = d_t - \frac{d_{tT} d'_{t-1}}{d'_{t-1} d'_{t-1}} d'_{t-1} \tag{10}$$

SD1: The above models mentioned till now impose a hard orthogonality constraint on the context vector(d'_t).So they proposed a relaxed version of the above model which uses a gating parameter. This gating parameter decides what fraction of the previous context vector should be subtracted from the current context vector . Following are the relevant equations :-

$$\gamma_t = W_g d_{t-1} + b_g$$

$$d'_t = d_t - \gamma_t \frac{d_{tT} d'_{t-1}}{d'_{t-1} d'_{t-1}} d'_{t-1} \tag{11}$$

Where $W_g \in R^{l4*l4}, b_g \in R^{l_4}, l4$ is the dimension of d_t as defined in equation (8).

SD2: The SD1 model only ensures that the current context vector is diverse w.r.t the previous context vector. It ignores all history before time step t 1. To account for the history, we treat successive context vectors as a sequence and use a modified LSTM cell to compute the new state at each time step . Following are the set of relevant equations:-

$$i_t = \sigma(W_i d_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f d_t + U_f h_{t-1} + b_f)$$

$$o_t = \sigma(W_o d_t + U_o h_{t-1} + b_o)$$

$$\hat{c}_t = \tanh(W_c d_t + U_c h_{t-1} + b_c)$$

$$c_t = i_t * \hat{c}_t + f_t * c_{t-1}$$

$$c^{diverse}_t = c_t - \frac{c^T_t c_{t-1}}{c_{t-1} c_{t-1}} c_{t-1} \tag{12}$$

$$g_t = \sigma(W_g d_t + U_g h_{t-1} + b_g)$$

$$h_t = d_t = o_t * g_t * tanh(c_t^{diverse}) \qquad (13)$$

Where $W_i, W_f, W_o, W_c \in R^{l_5 * l_4}, U_i, U_f, U_o, U_c \in R^{l_5 * l_4}, d_t$ is the l_4 dimensional output of equation (8) ; l_5 is number of hidden units in the LSTM cell.

4 Baseline Models

For the Baseline models they compared with two baseline diversity methods as used by Chen in [2]. The following text is the detailed explanation of them :-

M1: This model accumulates all the previous context vectors as $\sum_{j=1}^{t-1} d'_j$ and uses this history for computing a diverse context vector:

$$d'_t = tanh(W_c d_t - U_c \sum_{j=1}^{t-1} d'_j) \qquad (14)$$

where $W_c, U_c \in R^{l_4 * l_4}$ are diagonal matrices.

M2: In this model, in addition to computing a diverse context as described in the previous Equation , the attention weights at each time step are also forged to be diverse from the attention weights at the previous time step.They also maintain a history of attention weights and compute a diverse attention vector by subtracting the history from the current attention vector.

$$\alpha'_{t,i} = v_a^T tanh(W_a s_t + U_a d_t - b_a \sum_{j=1}^{t-1} \alpha'_{j,i})$$

Where $W_a \in R^{l_1 * l_1}, U_a \in R^{l_1 * l_4}, b_a, v_a \in R^{l_1}, l_1$ is the number of hidden units in the decoder .

5 Experiments

We evaluated our models on the dataset as proposed by [3].

Baseline: We experimented on baseline approaches by first using RNN based approaches , LSTM(again an RNN based approach) and finally implemented with multilayered bidirectional *GRU* with 4 layers.We used 80% of the data for training, 10% for validation and 10% for testing. *Note:*The results of our baselines outperformed the paper because of generating a mechanism to take output from the decoder directly and adding the facility of Bi-Direction.For Optimizer we used SGD intially and used ADAM optimiser after seeing improvements in results of ADAM we went with ADAM optimizer.We tried batch sizes of $8, 32, 64$.As batch size increased quality of summary generated decreases with relaxed training time as more padding needed to be imp laced. We achieved best result for batch size of 8.For Dropout we tried dropout with 0.3,0.4,0.5,0.7. After rigorous experimentation , we fixed the Dropout to 0.5 . In our model we also used pretrained glove embedding(840B300d version) of dimension 300 . We used different hidden size for encoder and decoder , the reason being the implementation of BiDirectional GRU in encoder , while single layer in decoder(due to . Thus encoder and decoder are having 256 and 512 as hidden state size respectively .We trained our model for 35 epoches in average.

Baseline plus: For models of Baseline plus all the hyper parameters were kept same as that in Baselines except some changes were forced at the time of training . The reason being that some models were not over fitting for standard number of epochs(35) (checked by looking at the trend of the validation loss) so we used 5 more epochs on some of them . For D2 we used a LSTM based approach . We tried multilayer approaches with 4 , 2 layers. But a single layer LSTM model worked the best .Then later we included the pretrained Glove embeddings on these models but almost resulted in similar performances as compared to random embeddings. For LSTM we tried DroupOut with the values 0.4 , 0.5 , 0.6 . For D1,D2 approaches , instead of adding the context vector as in equation(10),we added a linear layer and tried if it achieved some more improvements . But unfortunately , results saw no major improvement .The reason might be that the job of equation(10) is just to make previous context vector orthogonal to current context vector .

Experiments To be tried Despite multiple attempts to be perfect , perfection seems to be at the horizon, close yet very far.Thus there are some works that can be attempted to further refine our models. One of them will be to try with Transformer based approach to generate runtime embeddings . We will also wish to inculcate the input from the user regarding how long summary the user prefers .

6 Results and Detailed Analysis

Our all models outperformed the performance in comparison to the respective models of Nema's

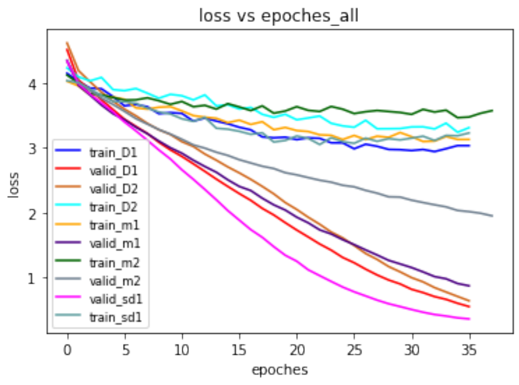
Approach [3] .

Reasons for Outperformance: Our model had a biderctional multilayer encoder for both the query and document whereas when it comes to hidden size of decoder , it is double of hidden size of query and document , whereas Nema used hidden state of same size , thus the concept building happened better . [3] .Also we took output directly from decoder instead of using state of decoder as done in paper.The other reasons include direct implementation of pytorch architecture vs making RNN's and LSTM from scratch .

Baseline Models A slight imporvement is seen in M1 when compared to corresponding baseline and Model M2 performed very badly.It may be that simultaneously adding a constraint on the context vectors as well as attention weights (as is indeed the case with M2) is a bit too aggressive and leads to poor performance.

Effect of Diversity models: As mentioned base line model suffers from the problem of repeating the same phrase/word while predicting the output . All the diversity models introduced in the (rows 4, 5, 6) give significant improvement over the non-diversity models. In particular, the modified LSTM based diversity model gives the best results. This is indeed very encouraging .It is because repetion of words is less in LSTM based approach which imporved results.

Loss Curves: Loss curves on all models showed that model almost all models are just fit and we used 35 epoches on all models.Figure Loss curve shows loss curve on all models.All models followed almost same trend in train loss vs test loss in some models loss decreased because of diversity addition.



:

Loss Curve

Model	Rouge-1	Rouge-2	Rouge-l
B1	52.5	31.2	51.9
M1	53.1	32.3	52.7
M2	39.7	12.9	39.04
D1	56.4	36.5	55.9
SD1	57.1	39.08	56.7
SD2	60.2	41.1	59.89

7 Conclusion

The approach used by [3] was incredible. The work provided a novel diversification mechanism based on successive orthogonalization . This lets us to (i) provide diverse context vectors at successive time steps (ii) pay attention to words repeatedly if need be later in the summary . We also discovered that their work was equally important to involve the creation of the dataset from debatepedia . We observed that the phenomenon of adding an attention mechanism on the query string as mentioned in the paper [3], does give significant improvements . The tweaks that we made involved using decoder to generate output , also we used Bi-directional RNN architecture and with an enhanced combination of layers . All these made our results overpower the results achieved in the parent paper [3].