Diversity driven Attention Model for Query-based Abstractive Summarization

Preksha Nema[†] **Mitesh M. Khapra**[†] **Anirban Laha**[*†] **Balaraman Ravindran**[†]
[†]Indian Institute of Technology Madras, India
[*] IBM Research India
{preksha,miteshk}@cse.iitm.ac.in
anirlaha@in.ibm.com ravi@cse.iitm.ac.in

Abstract

Abstractive summarization aims to generate a shorter version of the document covering all the salient points in a compact and coherent fashion. On the other hand, query-based summarization highlights those points that are relevant in the context of a given query. The encode-attend-decode paradigm has achieved notable success in machine translation, extractive summarization, dialog systems, etc. But it suffers from the drawback of generation of repeated phrases. In this work we propose a model for the query-based summarization task based on the encode-attend-decode paradigm with two key additions (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. In order to enable the testing of this model we introduce a new query-based summarization dataset building on debatepedia. Our experiments show that with these two additions the proposed model clearly outperforms vanilla encode-attend-decode models with a gain of 28% (absolute) in ROUGE-L scores.

1 Introduction

Over the past few years neural models based on the encode-attend-decode (Bahdanau et al., 2014) paradigm have shown great success in various natural language generation (NLG) tasks such as machine translation (Bahdanau et al., 2014), abstractive summarization ((Rush et al., 2015),(Nallapati et al., 2016)) dialog (Li et al., 2016), *etc*. One such NLG problem which has not received enough attention in the past is query based abstractive text summarization where the aim is to generate the summary of a document in the context of a query. In general, abstractive summarization, aims to cover all the salient points of a document in a compact and coherent fashion. On the other hand, query focused summarization highlights those points that are relevant in the context of the query. Thus given a document on "the super bowl", the query "How was the half-time show?", would result in a summary that would not cover the actual game itself.

Note that there has been some work on query based extractive summarization in the past where the aim is to simply extract the most salient sentence(s) from a document and treat these as a summary. There is no natural language generation involved. Since, we were interested in abstractive (as opposed to extractive) summarization we created a new dataset based on debatepedia. 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 of a sentence which is simply copied from the original document.

Using this dataset as a testbed, we focus on a recurring problem in models based on the encode-attend-decode paradigm. Specifically, it is observed that the summaries produced by such models contain repeated phrases. Table 1 shows a few such examples of summaries gener-

Document Snippet: The "natural death" alternative to euthanasia is not keeping someone alive via life support until they die on life support. That would, indeed, be unnatural. The natural alternative is, instead, to allow them to die off of life support. **Query**: Is euthanasia better than withdrawing life support (non-treatment)? **Ground Truth Summary**: The alternative to euthanasia is a natural death without life support. **Predicted Summary**: the large to euthanasia is a natural death **life life** use
Document Snippet: Legalizing same-sex marriage would also be a recognition of basic American principles, and would represent the culmination of our nation's commitment to equal rights. It is, some have said, the last major civil-rights milestone yet to be surpassed in our two-century struggle to attain the goals we set for this nation at its formation. **Query:** Is gay marriage a civil right? **Ground Truth Summary**: Gay marriage is a fundamental equal right. **Predicted Summary**: gay marriage is a appropriate **right right**

Table 1: Examples showing repeated words in the output of encoder-decoder models

ated by such a model when trained on this new dataset. This problem has also been reported by (Chen et al., 2016) in the context of summarization and by (Sankaran et al., 2016) in the context of machine translation.

We first provide an intuitive explanation for this problem and then propose a solution for alleviating it. 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. Each word is produced by feeding a new context vector to the decoder at each time step by attending to different parts of the document and query. If the decoder produces the same word or phrase repeatedly then it could mean that the context vectors fed to the decoder at these time steps are very similar.

We propose a model which explicitly prevents this by ensuring that successive context vectors are orthogonal to each other. Specifically, we subtract out any component that the current context vector has in the direction of the previous context vector. Notice that, we do not require the current context vector to be orthogonal to all previous context vectors but just its immediate predecessor. This enables the model to attend to words repeatedly if required later in the process. To account for the complete history (or all previous context vectors) we also propose an extension of this idea where we pass the sequence of context vectors through a LSTM (Hochreiter and Schmidhuber, 1997) and ensure that the current state produced by the LSTM is orthogonal to the history. At each time step, the state of the LSTM is then fed to the decoder to produce one word in the summary.

Our contributions can be summarized as follows: (i) We propose a new dataset for query based abstractive summarization and evaluate encode-attend-decode models on this dataset (ii) We study the problem of repeating phrases in NLG in the context of this dataset and propose two solutions for countering this problem. We show that our method outperforms a vanilla encoder-decoder model with a gain of 28% (absolute) in ROUGE-L score (iii) We also demonstrate that our method clearly outperforms a recent state of the art method proposed for handling the problem of repeating phrases with a gain of 7% (absolute) in ROUGE-L scores (iv) We do a qualitative analysis of the results and show that our model indeed produces outputs with fewer repetitions.

2 Related Work

Summarization has been studied in the context of text ((Mani, 2001), (Das and Martins, 2007), (Nenkova and McKeown, 2012)) as well as speech ((Zhu and Penn, 2006), (Zhu et al., 2009)). A vast majority of this work has focused on extractive summarization where the idea is to construct a summary by selecting the most relevant sentences from the document ((Neto et al., 2002), (Erkan and Radev, 2004), (Filippova and Altun, 2013), (Colmenares et al., 2015), (Riedhammer et al., 2010), (Ribeiro et al., 2013)). There has been some work on abstractive summarization in the context of DUC-2003 and DUC-2004 contests (Zajic et al.). We refer the reader to (Das and Martins, 2007) and (Nenkova and McKeown, 2012) for an excellent survey of

the field.

Recent research in abstractive summarization has focused on data driven neural models based on the encode-attend-decode paradigm (Bahdanau et al., 2014). For example, (Rush et al., 2015), report state of the art results on the GigaWord and DUC corpus using such a model. Similarly, the work of Lopyrev (2015) uses neural networks to generate news headline from short news stories. Chopra et al. (2016) extend the work of Rush et al. (2015) and report further improvements on the two datasets. Hu et al. (2015) introduced a dataset for Chinese short text summarization and evaluated a similar RNN encoder-decoder model on it.

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). Sankaran et al. (2016) study 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. Similarly, and more relevant to this work, Chen et al. (2016) propose 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. When evaluated on our dataset their method performs poorly. This could be because their method is very aggressive in dealing with the history (as explained later in the Experiments section). On the other hand, our method has a better way of handling history (by passing context vectors through an LSTM recurrent network) which gives us the flexibility to forget/retain some portions of the history and at the same time produce diverse context vectors at successive time steps.

We evaluate our method in the context of query based abstractive summarization - a problem which has received almost no attention in the past due to unavailability of datasets. We create a new dataset for this task and show that our method indeed produces better output by reducing the number of repeated phrases produced by encoder decoder models.

Average number of words per		
Document	Summary	Query
66.4	11.16	9.97

Table 2: Average length of documents/queries/summaries in the dataset

3 Dataset

As mentioned earlier, there are no existing datasets for query based abstractive summarization. We create such a dataset from Debatepedia an encyclopedia of pro and con arguments and quotes on critical debate topics. There are 663 debates in the corpus (we have considered only those debates which have at least one query with one document). These 663 debates belong to 53 overlapping categories such as Politics, Law, Crime, Environment, Health, Morality, Religion, etc. A given topic can belong to more than one category. For example, the topic "Eye for an Eye philosophy" belongs to both "Law" as well as "Morality". The average number of queries per debate is 5 and the average number of documents per query is 4. Please refer to the dataset url[1] for more details about number of debates per category.

For example, Figure 1 shows the queries associated with the topic "Algae Biofuel". It also lists the set of documents and an abstractive summary associated with each query. As is obvious from the example, the summary is an abstractive summary and not extracted directly from the document. We crawled 12695 such {query, document, summary} triples from debatepedia (these were all the triples that were available). Table 2 reports the average length of the query, summary and documents in this dataset.

We used 10 fold cross validation for all our experiments. Each fold uses 80% of the documents for training, 10% for validation and 10% for testing.

4 Proposed model

Given a query $\mathbf{q} = q_1, q_2, ..., q_k$ containing k words, a document $\mathbf{d} = d_1, d_2, ..., d_n$ containing n words, the task is to generate a contextual summary $\mathbf{y} = y_1, y_2, ..., y_m$ containing

[1] http://www.cse.iitm.ac.in/~miteshk/datasets/qbas.html

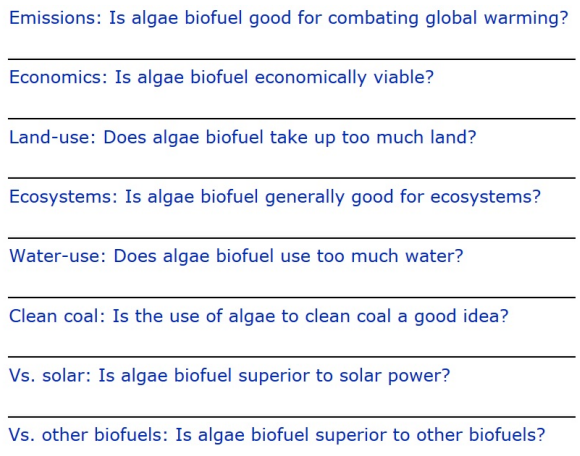

Figure 1: Queries associated with the topic "algae biofuel"

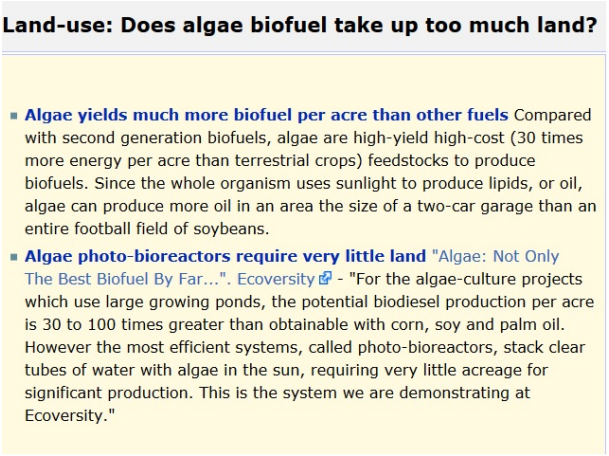

Figure 2: Documents and summaries for a given query

m words. This can be modeled as the problem of finding a $\mathbf{y}^*$ that maximizes the probability $p(\mathbf{y}|\mathbf{q}, \mathbf{d})$ which can be further decomposed as:

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

We now describe a way of modeling $p(y_t|y_1, ..., y_{t-1}, \mathbf{q}, \mathbf{d})$ using the neural encoder-attention-decoder paradigm. The proposed model contains the following components: (i) an encoder RNN for the query (ii) an encoder RNN for the document (iii) attention mechanism for the query (iv) attention mechanism for the document and (v) a decoder RNN. All the RNNs use a GRU cell.

Encoder for the query: We use a recurrent neural network with Gated Recurrent Units (GRU) for encoding the query. It reads the query $\mathbf{q} = q_1, q_2, ..., q_k$ from left to right and computes a hidden representation for each time-step as:

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

where $e(q_i) \in \mathbb{R}^d$ 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 $\mathbf{d} = d_1, d_2, ..., d_n$ from left to right and computes a hidden representation for each time-step as:

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

where $e(d_i) \in \mathbb{R}^d$ is the d-dimensional embedding of the document 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. We first describe the query attention model which assigns weights $\alpha_{t,i}^q$ to each word in the query at each decoder timestep using the following equations.

$$a_{t,i}^q = v_q^T \tanh(W_q s_t + U_q h_i^q) \quad (4)$$

$$\alpha_{t,i}^q = \frac{\exp(a_{t,i}^q)}{\sum_{j=1}^{k} \exp(a_{t,j}^q)} \quad (5)$$

where s_t is the current state of the decoder at time step t (we will see an exact formula for this soon). $W_q \in \mathbb{R}^{l_2 \times l_1}$, $U_q \in \mathbb{R}^{l_2 \times l_2}$, $v_q \in \mathbb{R}^{l_2}$, l_1 is the size of the decoder's hidden state, l_2 is both the size of h_i^q and also the size of the final query representation at time step t, which is computed as:

$$q_t = \sum_{i=1}^{k} \alpha_{t,i}^q h_i^q \quad (6)$$

Attention mechanism for the document : We now describe the document attention model which assigns weights to each word in the document using the following equations.

$$a_{t,i}^d = v_d^T \tanh(W_d s_t + U_d h_i^d + Z q_t) \quad (7)$$

$$\alpha_{t,i}^d = \frac{\exp(a_{t,i}^d)}{\sum_{j=1}^{n} \exp(a_{t,j}^d)}$$

where s_t is the current state of the decoder at time step t (we will see an exact formula for this

soon). $W_d \in \mathbb{R}^{l_4 \times l_1}$, $U_d \in \mathbb{R}^{l_4 \times l_4}$, $Z \in \mathbb{R}^{l_4 \times l_2}$, $v_d \in \mathbb{R}^{l_2}$, 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:

$$d_t = \sum_{i=1}^{n} \alpha_{t,i}^d h_i^d \tag{8}$$

Note that d_t now encodes the relevant information from the document as well as the query (see Equation (7)) at time step t. We refer to this as the *context vector* for the decoder.

Decoder: The hidden state of the decoder s_t at each time t is again computed using a GRU as follows:

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

where, y_{t-1} gives a distribution over the vocabulary words at timestep $t-1$ and is computed as:

$$y_t = \text{softmax}(W_o f(W_{dec} s_t + V_{dec} d_t)) \tag{10}$$

where $W_o \in \mathbb{R}^{N \times l_1}$, $W_{dec} \in \mathbb{R}^{l_1 \times l_1}$, $V_{dec} \in \mathbb{R}^{l_1 \times l_4}$, N is the vocabulary size, y_t is the final output of the model which defines a probability distribution over the output vocabulary. This is exactly the quantity defined in Equation (1) that we wanted to model $(p(y_t|y_1, ..., y_{t-1}, \mathbf{q}, \mathbf{d}))$. Further, note that, $e(y_{t-1})$ is the d-dimensional embedding of the word which has the highest probability under the distribution y_{t-1}. Also $[e(y_{t-1}), d_{t-1}]$ means a concatenation of the vectors $e(y_{t-1}), d_{t-1}$. We chose f to be the identity function.

The model as described above is an instantiation of the encoder-attention-decoder idea applied to query based abstractive summarization. As mentioned earlier (and demonstrated later through experiments), this model suffers from the problem of repeating the same phrase/word in the output. We now propose a new attention model which we refer to as diversity based attention model to address this problem.

4.1 Diversity based attention model

As hypothesized earlier, 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. We propose four models ($\mathbf{D_1}$, $\mathbf{D_2}$, $\mathbf{SD_1}$, $\mathbf{SD_2}$) to directly address this problem.

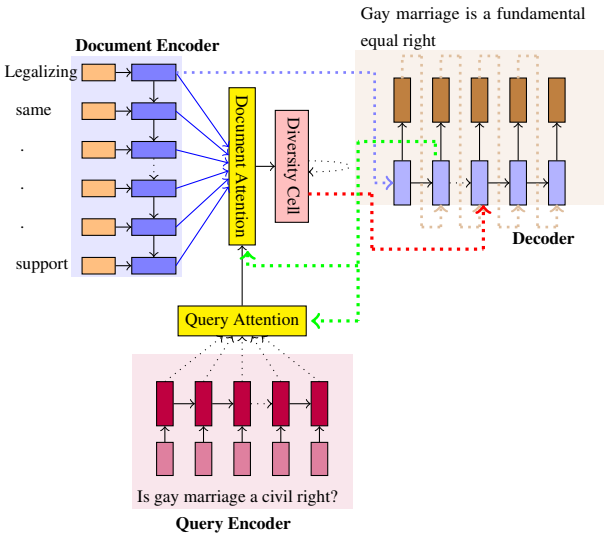

Figure 3: Proposed model for Query based Abstractive Summarization with (i) query encoder (ii) document encoder (iii) query attention model (iv) diversity based document attention model and (v) decoder. The green and red arrows show the connections for timestep 3 of the decoder.

$\mathbf{D_1}$: In this model, after computing d_t as described in Equation (8), we make it orthogonal to the context vector at time $t-1$:

$$d_t' = d_t - \frac{d_t^T d_{t-1}'}{d_{t-1}'^T d_{t-1}'} d_{t-1}' \tag{11}$$

$\mathbf{SD_1}$: The above model imposes a hard orthogonality constraint on the context vector(d_t'). We also propose 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 using the following equations:

$$\gamma_t = W_g d_{t-1} + b_g$$
$$d_t' = d_t - \gamma_t \frac{d_t^T d_{t-1}'}{d_{t-1}'^T d_{t-1}'} d_{t-1}'$$

where $W_g \in \mathbb{R}^{l_4 \times l_4}$, $b_g \in \mathbb{R}^{l_4}$, l_4 is the dimension of d_t as defined in equation (8).

$\mathbf{D_2}$: The above 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. Specifically, we use the following set of equations to compute a diverse context at time t:

$$
\begin{aligned}
i_t &= \sigma(W_i d_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f d_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o d_t + U_o h_{t-1} + b_o) \\
\hat{c}_t &= \tanh(W_c d_t + U_c h_{t-1} + b_c) \\
c_t &= i_t \odot \hat{c}_t + f_t \odot c_{t-1} \\
c_t^{diverse} &= c_t - \frac{{c_t}^T c_{t-1}}{c_{t-1}^T c_{t-1}} c_{t-1} \qquad (12) \\
h_t &= o_t \odot \tanh(c_t^{diverse}) \\
d_t^{'} &= h_t \qquad (13)
\end{aligned}
$$

where $W_i, W_f, W_o, W_c \in \mathbb{R}^{l_5 \times l_4}$, $U_i, U_f, U_o, U_c \in \mathbb{R}^{l_5 \times l_4}$, d_t is the l_4-dimensional output of Equation (8); l_5 is number of hidden units in the LSTM cell. This final $d_t^{'}$ from Equation (13) is then used in Equation (9). Note that Equation (12) ensures that state of the LSTM at time step t is orthogonal to the previous history. Figure 3 shows a pictorial representation of the model with a diversity LSTM cell.

$\mathbf{SD_2}$: This model again uses a relaxed version of the orthogonality constraint used in $\mathbf{D_2}$. Specifically, we define a gating parameter g_t and replace (12) above by (14) as define below:

$$
\begin{aligned}
g_t &= \sigma(W_g d_t + U_g h_{t-1} + b_o) \\
c_t^{diverse} &= c_t - g_t \frac{{c_t}^T c_{t-1}}{c_{t-1}^T c_{t-1}} c_{t-1} \qquad (14)
\end{aligned}
$$

where $W_g \in \mathbb{R}^{l_5 \times l_4}$, $U_g \in \mathbb{R}^{l_5 \times l_4}$

5 Baseline Methods

We compare with two recently proposed baseline diversity methods (Chen et al., 2016) as described below. Note that these methods were proposed in the context of abstractive summarization (not query based abstractive summarization) and we adapt them for the task of query based abstractive summarization. Below we just highlight the key differences from our model in computing the context vector $d_t^{'}$ passed to the decoder.

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

$$
d_t^{'} = \tanh(W_c d_t - U_c \sum_{j=1}^{t-1} d_j^{'}) \qquad (15)
$$

where $W_c, U_c \in \mathbb{R}^{l_4 \times l_4}$ are diagonal matrices. We then use this diversity driven context $d_t^{'}$ in Equation (9) and (10).

M2: In this model, in addition to computing a diverse context as described in Equation (15), the attention weights at each time step are also forced to be diverse from the attention weights at the previous time step.

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

where $W_a \in \mathbb{R}^{l_1 \times l_1}$, $U_a \in \mathbb{R}^{l_1 \times l_4}$, $b_a, v_a \in \mathbb{R}^{l_1}$, l_1 is the number of hidden units in the decoder GRU. Once again, they maintain a history of attention weights and compute a diverse attention vector by subtracting the history from the current attention vector.

6 Experimental Setup

We evaluate our models on the dataset described in section 3. Note that there are no prior baselines on query based abstractive summarization so we could only compare with different variations of the encoder decoder models as described above. Further, we compare our diversity based attention models with existing models for diversity by suitably adapting them to this problem as described earlier. Specifically, we compare the performance of the following models:

- **Vanilla e-a-d**: This is the vanilla encoder-attention-decoder model adapted to the problem of abstractive summarization. It contains the following components (i) document encoder (ii) document attention model (iii) decoder. It does not contain an encoder or attention model for the query. This helps us understand the importance of the query.
- $\mathbf{Query}_{enc}$: This model contains the query encoder in addition to the three components used in the vanilla model above. It does not contain any attention model for the query.

- $\mathbf{Query}_{att}$: This model contains the query attention model in addition to all the components in $Query_{enc}$.
- $\mathbf{D}_1$: The diversity attention model as described in Section 4.1.
- $\mathbf{D}_2$: The LSTM based diversity attention model as described in Section 4.1.
- $\mathbf{SD}_1$: The soft diversity attention model as described in Section 4.1
- $\mathbf{SD}_2$: The soft LSTM based diversity attention model as described in Section 4.1
- $\mathbf{B}_1$: Diversity cell in Figure3 is replaced by the basic LSTM cell (i.e. $c_t^{diverse} = c_t$ instead of using Equation (12). This helps us understand whether simply using an LSTM to track the history of context vectors (without imposing a diversity constraint) is sufficient.
- $\mathbf{M}_1$: The baseline model which operates on the context vector as described in Section 5.
- $\mathbf{M}_2$: The baseline model which operates on the attention weights in addition to the context vector as described in Section 5.

We used 80% of the data for training, 10% for validation and 10% for testing. We create 10 such folds and report the average Rouge-1, Rouge-2, Rouge-L scores across the 10 folds. The hyperparameters (batch size and GRU cell sizes) of all the models are tuned on the validation set. We tried the following batch sizes : 32, 64 and the following GRU cell sizes 200, 300, 400. We used Adam (Kingma and Ba, 2014) as the optimization algorithm with the initial learning rate set to 0.0004, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We used pre-trained publicly available Glove word embeddings[2] and fine-tuned them during training. The same word embeddings are used for the query words and the document words.

Table 3 summarizes the results of our experiments.

[2]http://nlp.stanford.edu/projects/glove/

Models	ROUGE-1	ROUGE-2	ROUGE-L
Vanilla e-a-d	13.73	2.06	12.84
$Query_{enc}$	20.87	3.39	19.38
$Query_{att}$	29.28	10.24	28.21
B1	23.18	6.46	22.03
M1	33.06	13.35	32.17
M2	18.42	4.47	17.45
D1	33.85	13.65	32.99
SD1	31.36	11.23	30.5
D2	38.12	16.76	37.31
SD2	**41.26**	**18.75**	**40.43**

Table 3: Performance on various models using full-length ROUGE metrics

7 Discussions

In this section, we discuss the results of the experiments reported in Table 3.

1. Effect of Query: Comparing rows 1 and 2 we observe that adding an encoder for the query and allowing it to influence the outputs of the decoder indeed improves the performance. This is expected as the query contains some keywords which could help in sharpening the focus of the summary.

2. Effect of Query attention model: Comparing rows 2 and 3 we observe that using an attention model to dynamically compute the query representation at each time step improves the results. This suggests that the attention model indeed learns to focus on relevant portions of the query at different time steps.

3. Effect of Diversity models: All the diversity models introduced in the paper (rows 7, 8, 9, 10) 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 and Table 4 shows some sample summaries comparing the performance of different models.

4. Comparison with baseline diversity models: The baseline diversity model M1 performs at par with our models D1 and SD1 but not as good as D2 and SD2. However, the model M2 performs very poorly. We believe 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 (although this needs further investigation).

5. Quantitative Analysis: In addition to the qualitative analysis reported in Table 4 we also did a quantitative analysis by counting the num-

Source: Although cannabis does indeed have some harmful effects, it is no more harmful than legal substances like alcohol and tobacco. As a matter of fact, research by the British Medical Association shows that nicotine is far more addictive than cannabis. Furthermore, the consumption of alcohol and the smoking of cigarettes cause more deaths per year than does the use of cannabis (e.g. through lung cancer, stomach ulcers, accidents caused by drunk driving etc.). The legalization of cannabis will remove an anomaly in the law whereby substances that are more dangerous than cannabis are legal whilst the possession and use of cannabis remains unlawful. **Query:** is marijuana harmless enough to be considered a medicine **G:** marijuana is no more harmful than tobacco and alcohol $Query_{attn}$: marijuana is no the **drug drug** for **tobacco and tobacco** **D1:** marijuana is no more harmful than tobacco and tobacco **SD1:** marijuana is more for evidence than tobacco and health **D2:** marijuana is no more harmful than tobacco and use **SD2:** marijuana is no more harmful than tobacco and alcohol
Source: Fuel cell critics point out that hydrogen is flammable, but so is gasoline. Unlike gasoline, which can pool up and burn for a long time, hydrogen dissipates rapidly. Gas tanks tend to be easily punctured, thin-walled containers, while the latest hydrogen tanks are made from Kevlar. Also, gaseous hydrogen isn't the only method of storage under consideration–BMW is looking at liquid storage while other researchers are looking at chemical compound storage, such as boron pellets. **Query:** safety are hydrogen fuel cell vehicles safe **G:** hydrogen in cars is less dangerous than gasoline $Query_{attn}$: hydrogen is **hydrogen hydrogen hydrogen** fuel energy **D1:**hydrogen in cars is less natural than gasoline **SD1:** hydrogen in cars is reduce risk than fuel **D2:** hydrogen in waste is less effective than gasoline **SD2:**hydrogen in cars is less dangerous than gasoline
Source: The basis of all animal rights should be the Golden Rule: we should treat them as we would wish them to treat us, were any other species in our dominant position. **Query:** do animals have rights that makes eating them inappropriate **G:** animals should be treated as we would want to be treated $Query_{att}$: animals should be **treated** as we would protect to be **treated** **D1:** animals should be **treated** as we most individual to be **treated** **SD1:** animals should be **treated** as we would physically to be treated **D2:** animals should be **treated** as we would illegal to be **treated** **SD2:** animals should be **treated** as those would want to be **treated**

Table 4: Summaries generated by different models. In general, we observed that the baseline models which do not use a diversity based attention model tend to produce more repetitions. Notice that the last example shows that our model is not very aggressive in dealing with the history and is able to produce valid repetitions (treated ... treated) when needed

ber of sentences containing repeated words generated by different models. Specifically for the 1268 test instances we counted the number of sentences containing repeated words as generated by different modes. Table 5 summarizes this analysis.

Model	Number
$\mathbf{Query}_{attn}$	498
$\mathbf{SD}_1$	352
$\mathbf{SD}_2$	344
$\mathbf{D}_1$	191
$\mathbf{D}_2$	179

Table 5: Average number of sentences with repeating words across 10 folds

8 Conclusion

In this work we proposed a query-based summarization method. The unique feature of

the model is a novel diversification mechanism based on successive orthogonalization. This gives us the flexibility to: (i) provide diverse context vectors at successive time steps and (ii) pay attention to words repeatedly if need be later in the summary (as opposed to existing models which aggressively delete the history). We also introduced a new data set and empirically verified we perform significantly better (gain of 28% (absolute) in ROUGE-L score) than applying a plain encode-attend-decode mechanism to this problem. We observe that adding an attention mechanism on the query string gives significant improvements. We also compare with a state of the art diversity model and outperform it by a good margin (gain of 7% (absolute) in ROUGE-L score). The diversification model proposed is general enough to apply to other NLG tasks with suitable modifications and we are currently working on extending this to dialog systems and general summarization.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. pages 2754–2760.

Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. 2016. Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of NAACL-HLT16* pages 93–98.

Carlos A Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. Heads: Headline generation as sequence prediction using an abstract feature-rich space. In *HLT-NAACL*. pages 133–142.

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.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22:457–479.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *EMNLP*. Citeseer, pages 1481–1491.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865* .

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155* .

Konstantin Lopyrev. 2015. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712* .

Inderjeet Mani. 2001. *Automatic summarization*, volume 3. John Benjamins Publishing.

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* .

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*, Springer, pages 43–76.

Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Brazilian Symposium on Artificial Intelligence*. Springer, pages 205–215.

Ricardo Ribeiro, Luís Marujo, David Martins de Matos, Joao P Neto, Anatole Gershman, and Jaime Carbonell. 2013. Self reinforcement for important passage retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 845–848.

Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. 2010. Long story short–global unsupervised models for keyphrase based meeting summarization. *Speech Communication* 52(10):801–815.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* .

Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, and Abe Ittycheriah. 2016. Temporal attention model for neural machine translation. *arXiv preprint arXiv:1608.02927* .

David Zajic, Bonnie Dorr, and Richard Schwartz. ???? Bbn/umd at duc-2004: Topiary.

Xiaodan Zhu and Gerald Penn. 2006. Comparing the roles of textual, acoustic and spoken-language features on spontaneous-conversation summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 197–200.

Xiaodan Zhu, Gerald Penn, and Frank Rudzicz. 2009. Summarizing multiple spoken documents: finding evidence from untranscribed audio. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, pages 549–557.