Diversity Driven Attention Model For Query-Based Abstraction Summarization

Abstract:

- Abstractive based summarization aims to generate a shorter version of document covering all the salient features of the document.
- Query based summarization aims to highlight those points in documents that are relevant to query.
- This project is based on encode-attend decode paradigm with two key additions
    - A query Attention model which learns to focus on different portions of the query at different time steps
    - A new diversity based attention model which aims to alleviate the problem of repeating phrases in the summary.

Introduction:

- Over the past few years neural models based on encode-attend-decode paradigm has shown great success in various Natural Language Generation tasks such as machine translation, abstractive summarization.In this project we attempt to generate an abstract summary in context of the query.
- Dataset is based on debatepedia. It contains triplet of the form(document, query, summary). 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.
- Next, we try to battle a recurring problem in attention models,it is observed that the summaries produced by such models contain repeated phrases.
- 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.
- Diversity models explicitly prevent this by ensuring successive context vectors are orthogonal to each other. For that subtract out any component that the current context vector has in the direction of the previous context vector.

- We do not need to have current vector orthogonal to all the previous context vectors but just its immediate predecessor. This enables the model to attend to words repeatedly if required later in the process.

Dataset :

- Dataset has been created from the debatepedia i.e an encyclopedia of pro and con arguments and quotes on critical debate topics.
- There are 663 debates in the corpus, considered only those debate that has at least one query with one document.
- These debates belong to several categories such as Politics, Law, Crime, Environment, Health, Morality, Religion, etc.

Proposed Model :

Given a query q = q1, q2, ..., qk containing k words, a document d = d1, d2, ..., dn containing n words, the task is to generate a contextual summary y = y1, y2, ..., ym containing m words.

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

Model above equation 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:

We use a recurrent neural network with Gated Recurrent Units (GRU) for encoding the query. It reads the query q = q1, q2, ..., qk from left to right and computes a hidden representation for each time-step as:

$$h_i^d = GRU_q(h_{i-1}^d , e(q_i))$$

where $e(q^i) \in R^d$ is the d-dimensional embedding of the query word q_i .

Similarly we can model the encoder for document

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.

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

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

Where s_t is the current state of the decoder at time step t.
$W_q \in R^{l2 \times l1}$, $U_q \in R^{l2 \times l2}$, $v_q \in R^{l2}$, l1 is the size of the decoder's hidden state, l2 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:

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

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.

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

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

where S_t is the current state of the decoder at time step t

$$d_t = \sum_{i=1}^{n} \alpha_{t,i}^{q} h_i^q$$

Decoder:

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

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

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

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

where Wo ∈ R $^{N \times l1}$, Wdec ∈ R $^{l1 \times l1}$, V_{dec} ∈ R $^{l1 \times l4}$, N is the vocabulary size, yt is the final output of the model which defines a probability distribution over the output vocabulary. This is exactly the quantity defined that we wanted to model (p(yt |y1, ..., yt−1, q, d)).

Diversity based Attention model:

If the decoder produce 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. propose four models (D1, D2, SD1, SD2) to directly address this problem.

D1: Next context vector(dt`) is the computed vector(dt) subtracted by the orthogonal projection of it on the previous context vector

$$d'_t = d_t - \frac{d_t^T d'_{t-1}}{d'^T_{t-1} d'_{t-1}} d'_{t-1}$$

SD1:

The above model imposes a hard orthogonality constraint on the context vector(d_t'). SD1 is basically the relaxed version of the D1, that 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:

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

$$d'_t = d_t - \gamma_t \frac{d_t^T d'_{t-1}}{d'^T_{t-1} d'_{t-1}} d'_{t-1}$$

D2 and SD2:

The above model ignores all history before time step t − 1. To account for the history, In model D2 we treat successive context vectors as a sequence and use a modified LSTM cell to compute the new state at each time step.

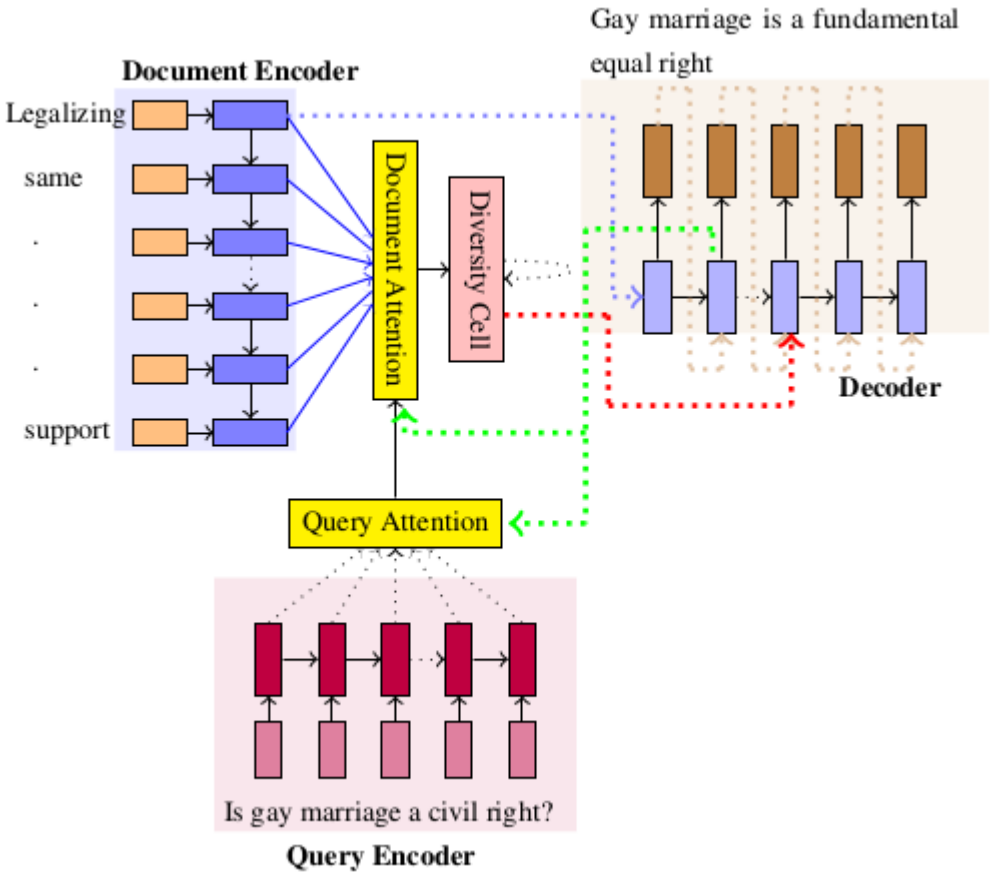
Below model(SD2) again uses a relaxed version of the orthogonality constraint used in D2.

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

$$c_t^{diverse} = c_t - g_t \frac{c_t^T c_{t-1}}{c_{t-1}^T c_{t-1}} c_{t-1}$$

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

Proposed Model - Figure



Results and Observations:

Main Hyper Parameters		
Parameter Name	**Accepted values**	**Default Value**
distraction_cell	"LSTM_soft, LSTM_hard, LSTM_subtract, "GRU_soft, GRU_hard, GRU_subtract"	LSTM_soft
diff_vocab	True , False	FALSE
is_distraction	True , False	TRUE
is_query_static	True , False	FALSE
is_dynamic	True , False	FALSE
is_bidir	True , False	TRUE
same_cell	True , False	FALSE
embedding_size	INT	300
hidden_size	INT	400
batch_size	INT	64
max_epochs	INT	50
learning_rate	FLOAT	0.004

Expt 1 : Varying hidden layer dimension hyperparameter				
hidden_size	**ROUGE-1**	**ROUGE-2**	**ROUGE-l**	**Test Loss**
200	22.4226804124	4.1509433962	17.7066946338	6.3045183897
400	26.4729620662	4.4689119171	18.8658865887	6.344333601
600	**30.2668938103**	4.2964035723	15.8352758353	6.3930488586
1000	29.0446471435	5.2257525084	15.7886815172	6.5624627829

Expt 2 : Varying learning rate parameter				
Learning rate	**ROUGE-1**	**ROUGE-2**	**ROUGE-l**	**Test Loss**
0.005	**40.2597402597**	14.3089430894	11.8075673778	7.0571278811
0.006	28.1189398836	5.539587144	17.4853259191	6.5896769047
0.009	30.0333704116	5.9392575928	16.7407832449	6.7305925608
0.01	18.685669042	0.8435136707	2.9491779364	25.0106981277

Conclusion:

- Query attention mechanism produces better summaries rather than just applying document attention on top relevant documents.
- Diversity model is effective is reducing the repeat words produced in the summary.
- LSTM driven diversity model is more suited for its flexible range of previous history considered and fine tuning it provided in the subtraction.