Improving Seq2Seq RNN model for Text Summarization

Rahul Chittimalla
CDS
Sr no: 14585
rahulc@iisc.ac.in

Vipul Kumar Rathore
CDS
Sr no: 14754
vipulrathore@iisc.ac.in

Dinesh Kumar
CDS
Sr no: 14428
dineshk@iisc.ac.in

Abstract

Traditionally, summarization has been approached through *extractive* methods. However, they have produced limited results. More recently, neural sequence-to-sequence models for *abstractive* text summarization have shown more promise. In this paper, we explore current state-of-the-art architectures and re-implement them from scratch. We implement the base paper given in (Nallapati et al., 2016a). The architecture in the paper is made of bi-directional GRU-RNN with Bahdanau(Bahdanau et al., 2014) attention mechanism. We implement LSTM as the RNN and change the attention mechanism to Luong(Luong et al., 2015) attention and do a comparative study.

1 Introduction

There is enormous amount of textual material like web pages, news articles, blogs etc. Textual information in the form of digital documents quickly accumulates to huge amounts of data. Most of this large volume of documents is unstructured: it is unrestricted and has not been organized into traditional databases. Processing documents is therefore a perfunctory task, mostly due to the lack of standards(Mani and Maybury, 1999). We cannot possibly create summaries of all of the text manually; there is a great need for automatic methods. Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks). More recently, researchers have been taken more of an abstractive approach, which is a bottom-up method that captures semantics that are not necessarily in the original text.

2 Problem

In the text summarization problem, we want to map an input sequence of M word vectors, $x_1, ..., x_m$ from a fixed vocabulary V to an output of another sequence of lenght $N < M$ word vectors.

3 Related Work

In (Rush et al., 2015), they approach the problem by utilizing a local attention-based model that generates each word of the summary conditioned on the input sentence. In (Nallapati et al., 2016b), they approach this summarization task using sequence-to-sequence RNNs with several optimizations to deal with of-of-vocabulary(OOV) tokens. They introduce a pointer model that uses a 'switch' per time-step that determines whether to use a word from the vocabulary or to point back to a particular word in the input.

4 Models

In this section, we describe our (1) basic sequence-to-sequence encode decoder attention model as our baseline, (2) ConceptNet Numberbatch word embeddings, (3) bi-directional LSTM model for the encoder, and finally (4) Luong attention mechanism

4.1 Sequence-to-sequence model

Our baseline model follows the same model presented in (Nallapati et al., 2016a) that uses GRU-RNN cells for both encoding and decoding. In the encoder, a token of the article x_i is fed at each time-step t. All embeddings were initialized from a random uniform distribution and learned during training. Any sequence longer than the max encoding length was truncated to fit the encoder, and short sequences were padded to fit. In the attention model, at every decoder state we generate a

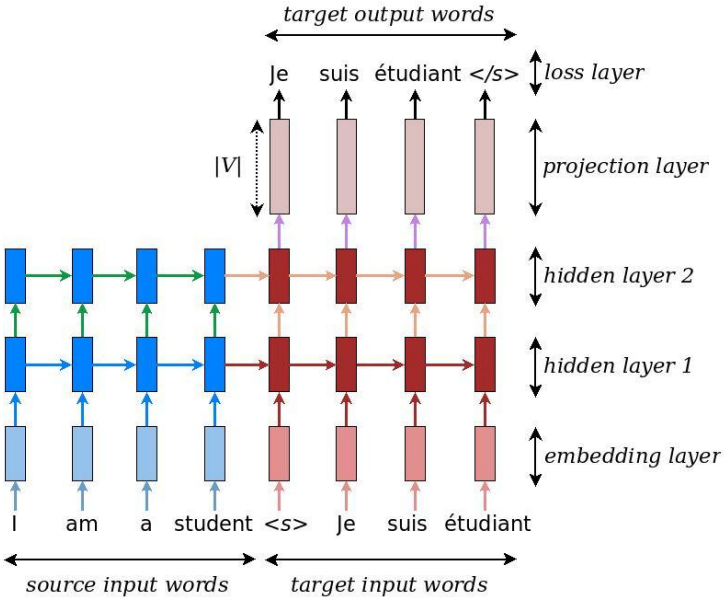

Figure 1: Basic Sequence-to-sequence model

context vectore based on all the hidden states generated in the encoding stage. The context vector at a given decode stage i can be computed with encoding hidden state h_j and the previous decoder hidden state s_{i-1} with a weight matrix W_a having dimensions D_h x D_h,

$$a_j = softmax(h_j^T W_a s_{j-1})$$

$$c_i = \sum_{j=1}^{n} a_j h_j$$

In the above equation, n is the encoding length used. From the context vector, the previous hidden state, and the input to the current decoder RNN, the new output word distribution can be generated.

4.2 ConceptNet Numberbatch word embeddings

ConceptNet Numberbatch(Speer et al., 2016) consists of state-of-the-art semantic vectors (also known as word embeddings) that can be used directly as a representation of word meanings or as a starting point for further machine learning. ConceptNet Numberbatch is part of the ConceptNet open data project. ConceptNet provides lots of ways to compute with word meanings, one of which is word embeddings. It is built using an ensemble that combines data from ConceptNet, word2vec, GloVe, and OpenSubtitles 2016, using a variation on retrofitting. We use these word embeddings to feed in to our bi-directional LSTM model which is described in the next section.

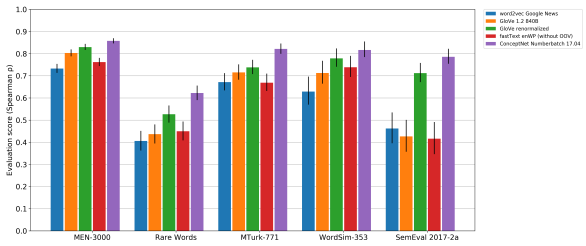

Figure 2: ConceptNet Numberbatch

4.3 bi-directional LSTM model for encoder

In our model, the encoder uses a bi-directional LSTM instead of GRU-RNN which was used in the baseline. It concatenates hidden states at each time step from both the previous and next word, capturing semantics from both sides of each word. The number of hidden units in each encoder LSTM was half of the number of hidden units in the output, so that in the same number of units the forward and backward state can be concatenated.

4.4 Luong attention mechanism

In this mechanism, there are 2 approaches - a global approach, in which all the source words are attended and a local one, whereby only a subset of source words are considered at one time. We make use of local attention in our model, since the global one is computationally expensive and impractical for longer sequences, such as paragraphs or documents.

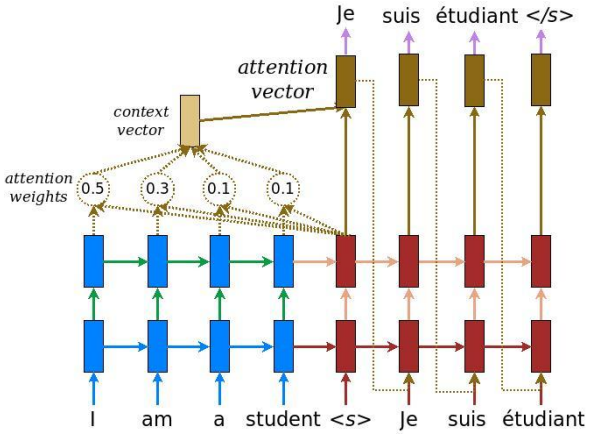

Figure 3: Sequence-to-sequence model with attention

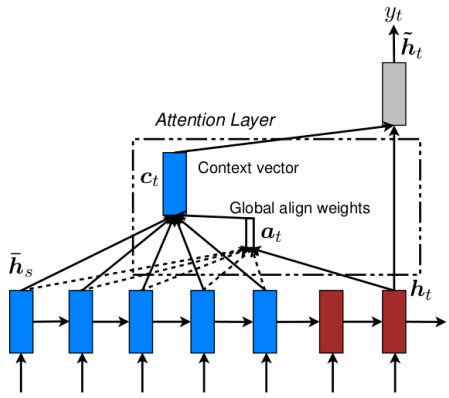

Figure 4: Global Mechanism.

5 Dataset and Cross-Validation

We are making the splits as train:development:test :: 80%:10%:10%. The development set is used for hyper-parameter tuning and test set is used for generating the summaries.
We are comparing the predicted summary with original summary manuall for each of the models.
We are conducting our experiments separately for each of the following 2 datasets -

- Amazon reviews Kaggle dataset - https://www.kaggle.com/snap/amazon-fine-food-reviews
- CNN news dataset - https://cs.nyu.edu/~kcho/DMQA/

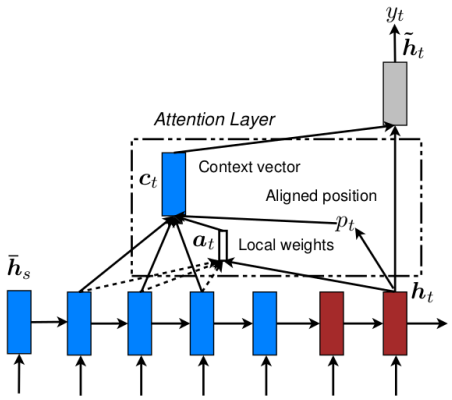

Figure 5: Local Mechanism.

Evaluation Metric We can use the **ROUGE-1 score** as our evaluation metric which is given as :-

$$ROUGE{-}1 = \frac{\textit{Number of overlapping words}}{\textit{Total number of words in reference summary}}$$

We calculated rouge score for our generated summaries but this might be deceptive for many cases as mentioned in next section.

6 Experiments and Results

- Model 1 - **LSTM + Bahdanau attention on CNN** -
  The plot of training accuracy over the iterations is as follows -

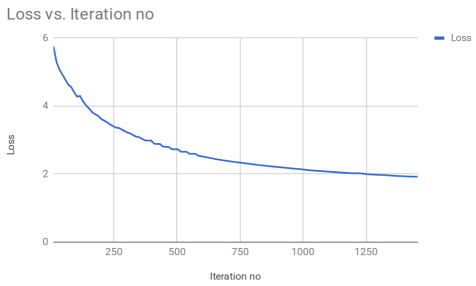

Figure 6: LSTM + Bahdanau on CNN dataset

The training loss, which is cross entropy loss in our case, reaches around 1.9 in around 1500 iterations. Some of the summaries generated are as follows -

1. **Text** - allan levene really really wants serve congressso much plans run four states georgia minnesota michigan hawaii.
   Predicted summary - allan levene

minnesota allan running after helping
Original summary - allan levene wants to run for congress in georgia minnesota michigan and hawaii
ROUGE-1 = 23.07 %

2. **Text** - ht making movie watch oscar awards approach next monthviola davis best actress trophy octavia spencer given best supporting actress honor women portrayed maids
Predicted summary - best actress wins best picture oscar oscar oscar
Original summary - new the help wins best cast best actress best supporting actress at sag awards
ROUGE-1 = 42.86 %

3. **Text** - seeing texas gov rick perrys lackluster performance debates accompanying drop polls pundits conclude would take force nature save campaign fortunately perry force exist name sarah palina dose palin power would much revitalize perrys chances win republican presidential nomination
Predicted summary - rick perrys says the republican showed showed lee be losing
Original summary - sarah palins endorsement could save rick perrys presidential campaign shayne lee says
ROUGE-1 = 33.33 %
Since, we are not getting quite good results on CNN dataset, we dropped the idea of going further with it and opted for Amazon Reviews dataset.

- Model 2 - **LSTM + Bahdanau attention on Amazon reviews** -
Some of the summaries generated are as follows -

1. **Text** - makes great vegan substitute coffee creamer calories takes less evens us husband insists using half n half coffee use although use regular powdered coffee creamer compared half n half really much higher calories vegan also used reconstituted cooking good results recommend product anyone following vegan diet.
Predicted summary - great for diabetics
Original summary - my picky husband loves it in his coffee
ROUGE-1 = 0. However the generated summary is quite good when tested manually.This is one of the cases where ROUGE evaluation metric fails miserably.

2. **Text** - subscription item 10 cheaper costco plus taxes ship house cannot get better
Predicted summary -the best afford have out there lol lol
Original summary - cheaper than costco
ROUGE-1 = 0, but predicted summary seems good.

3. **Text** - three rescued cats feed regular food course want treat eat whiskas savory salmon tried expensive treats pretty near every type brand come whiskas savory salmon rate five star treat
Predicted summary - my cat loves these
Original summary - my cat s favorite treat
ROUGE-1 = 50 %

4. **Text** - using coffee years brother lives texas uses got wife turned available stores minnesota found amazon purchase running low give try probably hooked well cheaper fancy grinds store mild great taste wife even like regular coffee longer
Predicted summary - great taste
Original summary - great coffee
ROUGE-1 = 50 %

- Model 3 - **LSTM + Lounge attention on Amazon reviews** -
Some of the summaries generated are as follows -

1. **Text** - care bad taste many brands k cup cider far better one purchase one
Predicted summary - better than expected

Original summary - it has a bad after taste
ROUGE-1 = 0, and even the predicted summary is not that great.

2. **Text** - ordered 20 candy tins party french themed guest honor requested gifts favorite girl tins showed good time half damaged cases unusable put severely dinged tins gift bags embarrassing explain party worked hard elegant detail would send back time instead apologize half guests damper would never give vendor second chance
Predicted summary -great product
Original summary - very damaged goods
ROUGE-1 = 0, and the predicted summary is also not good for this model.
3. **Text** - got cooked grill good think better frozen ones found grocery store though compare prices see better deal either way glad got one fish like mild texture good holds together well grill use
Predicted summary - better than expected
Original summary - delicious
ROUGE-1 = 0, but predicted summary sounds good.

- Model 4 - **Baseline model (GRU + Bahdanau attention) on Amazon reviews** -
Some of the summaries generated are as follows -

1. **Text** - becoming addicted mallow magic popcorn jolly time unable find stores amazon wonderful products really hard find reasonable price
Predicted summary - great popcorn
Original summary -love this can t find it in stores thanks amazon
ROUGE-1 = 0, but summary seems quite good.
2. **Text** - makes great vegan substitute coffee creamer calories takes less evens us husband insists using half n half coffee use although use regular powdered coffee creamer compared half n half really much higher calories vegan also used reconstituted cooking good results recommend product anyone following vegan diet
Predicted summary -great product
Original summary - my picky husband loves it in his coffee
ROUGE-1 = 0, but the predicted summary seems ok.
3. **Text** - subscription item 10 cheaper costco plus taxes ship house cannot get better
Predicted summary - love these
Original summary - my cat s favorite treat
ROUGE-1 = 0, and the predicted summary is not as good as in case of Model-1.
4. **Text** - hesitant try soup lentils appealing soup amazing filling healthy delicious
Predicted summary - redskin soup
Original summary - great soup
ROUGE-1 = 50 %

The plot of Training loss with iteration number for the Amazon Reviews Dataset is as shown below :-

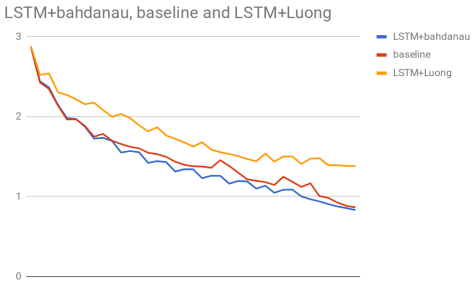

Figure 7: Comparison of 3 models on Amazon Reviews dataset

7 Inference

From the above summaries and plot for loss function, we conclude that we beat the baseline by replacing GRU encoder with bidirectional LSTM encoder. However, by changing the attention mechanism from Bahdanau to Lounge, we can't beat the baseline.Thus LSTM based encoder-decoder models can perform remarkably well for longer sequences such as paragraphs and documents for summarization tasks, probability due to their ability to capture long-distance dependencies in the sequences.

8 Future Work

In extension to this model, we can incorporate pointer-generator networks on top of this model to deal with words that come first time in test data by copying words from source text via pointing which aids accurate reproduction of information, while retaining the ability to produce novel words through the generator.

References

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Inderjeet Mani and Mark T Maybury. 1999. *Advances in automatic text summarization*. MIT press.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016a. Sequence-to-sequence rnns for text summarization.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016b. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

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

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975.