A Hierarchical Model for Text Autosummarization

Tapan Sahni, CS13B1030
Hrishikesh Vaidya, CS13B1035
Akilesh B, CS13B1042

Indian Institute of Technology, Hyderabad

November 17, 2016

Introduction

- Summarization is a central task in natural language processing.
- Most summarization systems are *extractive* methods. However, *extractive* methods are limited by their nature, as summaries do not essentially come from the source.
- On the other hand, *abstractive* methods generate summaries from the source, although they may not appear as part of the original text.

Intro contd . . .

- Recent LSTM models (Hochreiter and Schmidhuber, 1997) have shown powerful results on generating meaningful and grammatical sentences in sequence generation tasks like machine translation (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015) or parsing (Vinyals et al., 2014).
- This performance is at least partially attributable to the ability of these systems to capture local compositionally: the way neighboring words are combined semantically and syntactically to form meanings that they wish to express.

Our work

- In this project, our target is to use an encoder-decoder model to summarize a news article.
- Specifically, a hierarchical LSTM model is used as an encoder, in which the representations of sentences are learned by a LSTM model whose inputs are words.
- The representation of the document is learned by another LSTM model whose inputs are sentences representations.
- A normal LSTM is used as a decoder.

Model 1 : Standard LSTM

- The whole input and output are treated as one sequence of tokens.
- Following Sutskever et al. (2014) and Bahdanau et al. (2014), we trained an autoencoder that first maps input documents into vector representations from a $LSTM_{encode}$ and then reconstructs inputs by predicting tokens within the document sequentially from a $LSTM_{decode}$.
- Two separate LSTMs are implemented for encoding and decoding with no sentence structures considered.

Standard Sequence to Sequence Model

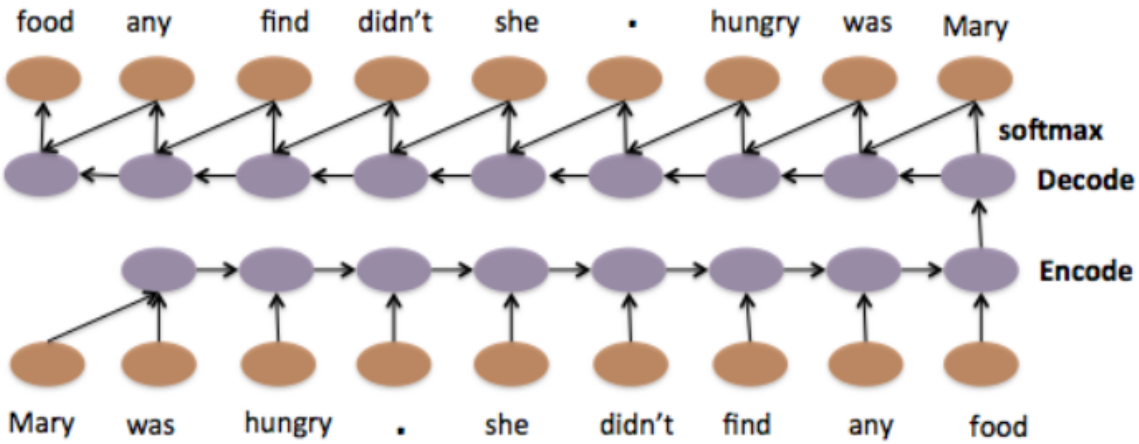

Figure 1: Standard Sequence to Sequence Model.

Model 2 : Hierarchical LSTM

Encoder We first obtain representation vectors at the sentence level by putting one layer of LSTM (denoted as $LSTM^{word}_{encode}$) on top of its containing words:

$$h^w_t(enc) = LSTM^{word}_{encode}(e^w_t, h^w_{t-1}(enc)) \quad (1)$$

The vector output at the ending time-step is used to represent the entire sentence as $e_s = h^w_{end_s}$

Hierarchical LSTM contd. . . .

To build representation e_D for the current document/paragraph D, another layer of LSTM (denoted as $LSTM_{encode}^{sentence}$) is placed on top of all sentences, computing representations sequentially for each time step:

$$h_t^s(enc) = LSTM_{encode}^{sentence}(e_t^s, h_{t-1}^s(enc)) \tag{2}$$

Representation $e_{end_D}^s$ computed at the final time step is used to represent the entire document: $e_D = h_{end_D}^s$

Thus one LSTM operates at the token level, leading to the acquisition of sentence-level representations that are then used as inputs into the second LSTM that acquires document-level representations, in a hierarchical structure.

Hierarchical LSTM contd. . . .

Decoder As with encoding, the decoding algorithm operates on a hierarchical structure with two layers of LSTMs. LSTM outputs at sentence level for time step t are obtained by:

$$h_t^s(dec) = LSTM_{decode}^{sentence}(e_t^s, h_{t-1}^s(dec)) \tag{3}$$

The initial time step $h_0^s(d) = e_D$, the end-to-end output from the encoding procedure, $h_t^s(d)$ is used as the original input into the $LSTM_{decode}^{word}$ for subsequently predicting tokens within sentence $t + 1$.

$LSTM_{decode}^{word}$ predicts tokens at each position sequentially, the embedding of which is then combined with earlier hidden vectors for the next time step prediction until the end_s token is predicted.

Hierarchical Sequence to Sequence Model

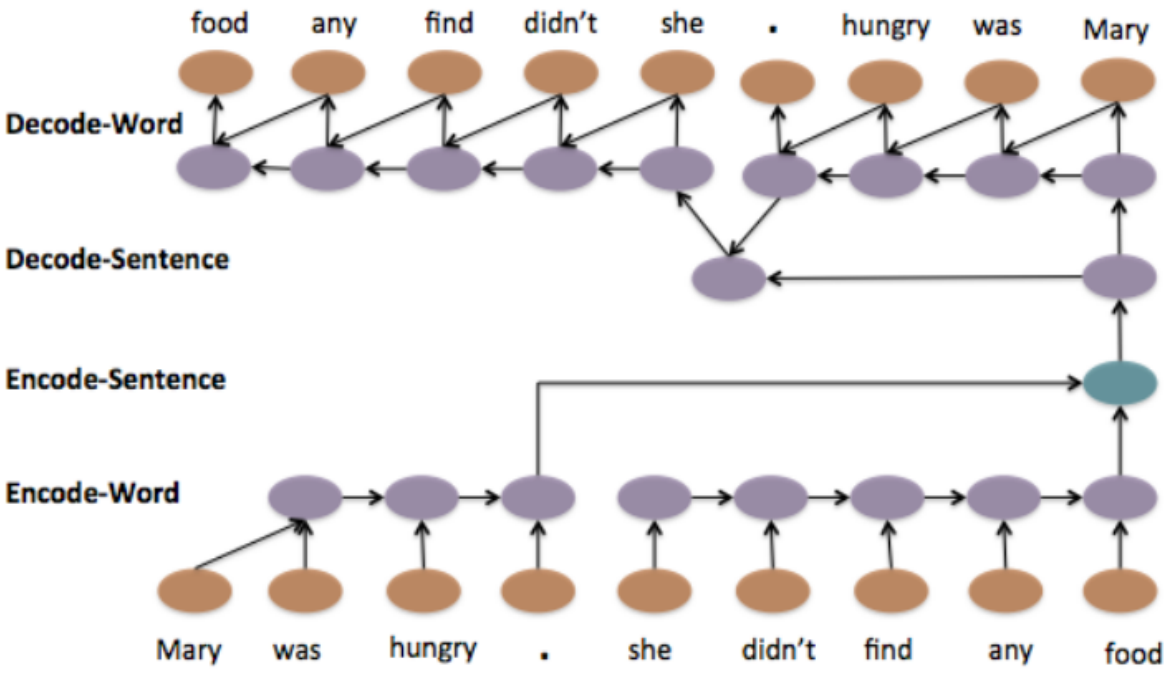

Figure 2: Hierarchical Sequence to Sequence Model.

Dataset

The Signal Media One-Million News Articles Dataset is used for training and testing. It contains 1 million articles that are mainly English, but they also include non-English and multi-lingual articles.
Each article in the dataset has the following fields.

- **id**: a unique identifier for the article.
- **title**: the title of the article.
- **content**: the textual content of the article (which occasionally contained HTML and JavaScript content).
- **source**: the name of the article source (e.g. Reuters).
- **published**: the publication date of the article.
- **media-type**: either "News" or "Blog".

Data Pre-processing

- Minimal pre-processing such as lower-casing, separating on the basis of punctuation marks are applied.
- All the punctuations are removed in order to reduce the vocabulary size.
- We remove the articles with more than 20% label the tokens which occur less than 10 times in our corpus.
- The average number of words in an article is 122 .

Data Pre-processing contd. . . .

- For every article, cosine similarity of each sentence present in it, is computed with the corresponding title. We retain only the sentences with top 5 cosine similarity scores per article.
- To make sure our system is actually summarizing text and not just returning the lead paragraphs, we shuffle the sentences before summarizing.
- One of the biggest elements of practically implementing our proposed solutions was the preprocessing step. This primarily included pruning out bad examples such as articles with no content or summaries with no content, but also included handling non-unicode characters.

Constructing feature vector and training

- We initially trained a Word2vec model on our whole corpus, which is a two-layer neural net that generates word embeddings. We found out that the results were not good.
- We used a one-hot encoding for our feature vector. This means the size of feature vector will be equal to the vocabulary.
- The standard LSTM encoder decoder model is implemented in Keras / Theano.
- Adagrad/Adadelta is used as optimizer for training.

Evaluation Metric

We use ROUGE-1,ROUGE-2 scores on the dataset. **ROUGE** is a recall-oriented measure widely used in the summarization literature. It measures the n-gram recall between the candidate text and the reference text(s).

$$ROUGE_n = \frac{\sum_{S \in \{RefSummaries\}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in \{RefSummaries\}} \sum_{gram_n \in S} count(gram_n)} \quad (4)$$

where $count_{match}$ denotes the number of n-grams co-occurring in the input and output.

Evaluation Metric . . .

BLEU Purely measuring recall will inappropriately reward long outputs. BLEU is designed to address such an issue by emphasizing precision. n-gram precision scores for our situation are given by:

$$precision_n = \frac{\sum_{gram_n \in output} count_{match}(gram_n)}{\sum_{gram_n \in output} count(gram_n)} \quad (5)$$

BLEU first compute the geometric average of the modified n-gram precisions, p_n, using n-grams up to length N and positive weights w_n summing to one. It then combines the average logarithm of precision scores with exceeded length penalization.

Evaluation Metric Comparision

BLEU How much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries. It measures precision.

ROGUE How much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries. It meassures recall.

Training details

Table: Hyperparameter list

Hyperparameter	Value
Hidden layer size	500/1000
Batch size	128
Learning rate	0.1
Optimizer	Adagrad/Adadelta

- The training was performed on a Tesla K-20 GPU with 6GB RAM.

Standard LSTM examples

Generated:
the new remote control about thicker than ray previous changed ipad apps of case girl s the good thing
Original:
the new remote control is thicker than the previous one and in this case that s a good thing
Generated: (Bad Summary)
to we've difference and diversity we've been by
Original:
where we've succeeded, its because we've been different.

Hierarchical LSTM examples

Generated:
if we spread malicious gossip we our poisoning words own peace demoralise sabotaging situations own influence
Original:
if we spread malicious gossip we are poisoning our own future and sabotaging our own influence with others
Generated:
pressing down on the screen starts different actions depending fathers how much force healing exerted
Original:
pressing down on the screen starts different actions depending on how much force is exerted

Hierarchical LSTM examples . . .

Generated:
the misunderstood tried searching i action movies the comedy movies the the remote control
Original:
i also tried searching for action movies and comedy movies with the remote control s microphone
Generated:
bowling ball on third 3rd side with ease redzone left texas went to chris nutall
Original:
bowling ball on third 3rd and in the redzone texas state went to chris nutall senior running back

Hierarchical LSTM summary eg.1

Original Text:(It's preprocessed text)
here are some first impressions of the iphone 3d touch capability the iphone signature new feature pressing down on the screen starts different actions depending on how much force is exerted when you press down hard the device responds with a tapping sensation i found the new iphones pretty straightforward to use the ipad pro is capable of running two full size ipad apps on the screen at the same time the big screen ipad also felt comfortable to hold it weighs about grams which is the same weight as the very first ipad the apple tv is significantly different from past models the software interface has a white background and vivid colours similar to apple tv music app the remote also includes motion sensors for gaming the new remote control is thicker than the previous one and in this case that is a good thing

Hierarchical LSTM summary eg.1. . .

Summary:
here are some impressions of iphone products capability pressing down on the screen starts different actions depending on how much force is exerted the big screen ipad capable felt comfortable two hold same weight as the very first ipad the tv software which is significantly same vivid colours models the new remote control is thicker than the previous

Hierarchical LSTM summary eg.2

Original Text:(It's preprocessed text)
an email chain between former secretary of state hillary clinton and then commander of united states is raising questions the chain included clinton and petraeus getting acquainted as well as some personnel matters according to the officials the chain begins on january and ends on feb and contains less than 10 emails in total between clinton and petraeus the official said these emails are now in our possession and will be subject to freedom of information act requests furthermore we asked the ig to incorporate this matter into the review secretary kerry requested in march we have also informed congress of this matter the official said the emails were found because they have now been digitized and are easier to search clinton turned over to state her work related emails from the server and kept emails she deemed personal the fbi is looking into whether any classified materials were mishandled

Hierarchical LSTM summary eg.2 . . .

Summary:
email chain between former secretary of state hillary clinton and the commander of united states the chain begins on january and ends and feb the contains less than emails between clinton and petraeus the official said furthermore we asked the ig to incorporate related matter into the review the fbi over looking into whether any classified materials were mishandled

Results

Table: Results for Hierarchical LSTM

No of articles	ROUGE-1 pre	ROUGE-1 rec	ROUGE-1 F1
100	0.1769	0.7561	0.2931
200	0.2372	0.8143	0.3835
500	0.3513	0.7713	0.5104

Table: Results for Standard LSTM

No of articles	ROUGE-1 pre	ROUGE-1 rec	ROUGE-1 F1
500	0.1267	0.4113	0.1539

Results . . .

Table: Results for Hierarchical LSTM

No of articles	ROUGE-2 pre	ROUGE-2 rec	ROUGE-2 F1
100	0.1131	0.4997	0.1969
200	0.1428	0.6014	0.2309
500	0.2828	0.5945	0.4222

Table: Results for Standard LSTM

No of articles	ROUGE-2 pre	ROUGE-2 rec	ROUGE-2 F1
500	0.0892	0.2591	0.1004

Results . . .

Table: Results for Hierarchical LSTM

No of articles	ROUGE-3 pre	ROUGE-3 rec	ROUGE-3 F1	BLEU
100	0.039	0.2043	0.0079	0.1145
200	0.065	0.2804	0.1075	0.2089
500	0.1565	0.4465	0.3876	0.3702

Table: Results for Standard LSTM

No of articles	ROUGE-3 pre	ROUGE-3 rec	ROUGE-3 F1	BLEU
500	0.002	0.1647	0.0059	0.007

Challenges faced

- Computational constraints - GPU.
- Huge size of vocabulary due to one-hot encoding.
- Hyperparameter optimizations which ensure error function converges to an appropriate local minima.
- Pruning out bad examples such as articles with no content or summaries.

References

 A Hierarchical Neural Autoencoder for Paragraphs and Documents. Jiwei et al.

 Sequence to sequence learning with Neural Networks. Sutskever et al.

 http://research.signalmedia.co/newsir16/signal-dataset.html