Neural Text Summarization

SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents

Xin SUN

IWR & CL

13. Nov 2019

Contents

- Text summarization background
- GRU simplified introduction
- SummaRuNNer
- Construction of Training data
- Experiment & Setting & Evaluation

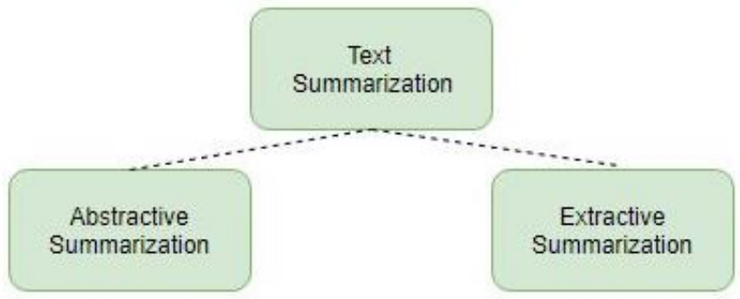
• Text summarization background

Definition:

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning of text.

Approach:

T**wo main types** of text summarization.

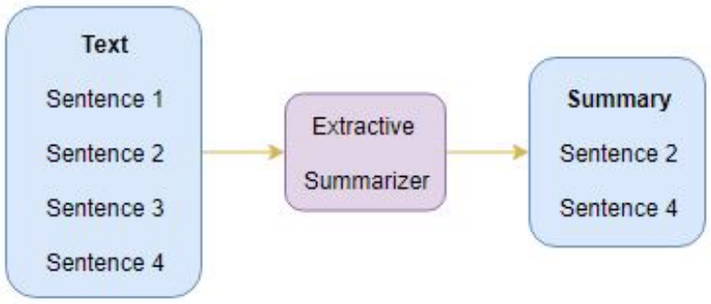


Extraction-based summarization.

Example:

Source text: ***Joseph and Mary*** *rode on a donkey to* ***attend*** *the annual* ***event*** *in* ***Jerusalem****. In the city,* ***Mary*** *gave* ***birth*** *to a child named* ***Jesus****.*

Extractive summary: *Joseph and Mary attend event Jerusalem. Mary birth Jesus.*

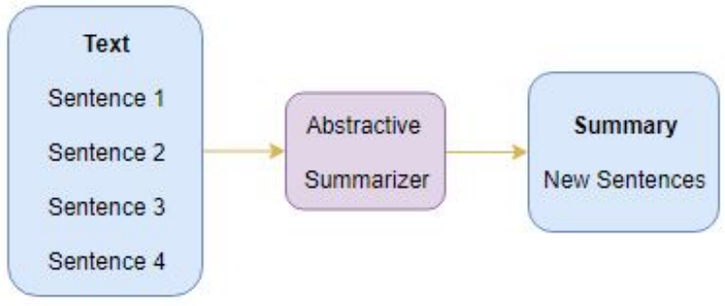


Abstraction-based summarization.

Example:

Source text: ***Joseph and Mary*** *rode on a donkey to* ***attend*** *the annual* ***event*** *in* ***Jerusalem****. In the city,* ***Mary*** *gave* ***birth*** *to a child named* ***Jesus****.*

Abstractive summary: *Joseph and Mary came to Jerusalem where Jesus was born.*



• Text summarization background

How does a simple extractive summarization algorithm works?

1. Pre-processing
• Tokenization, lemmatization, stop-words, non-sense words, etc.

2. Word, sentence representation
• Represent Words and sentences as vectors
Bag Of Words. / LDA. / Word Embedding. / TF-IDF-weighting.

3. Sorting
• **Sorting based on graph**
Using each sentence of the document as a node, the similarity between the sentences is used as the weight of edges to construct the graph model, and then use traditional sorting algorithm to sort each sentence.
—TextRank and LexRank.

• **Sorting based on features**
Features used here include:
1) The length of a sentence
2) The position of the sentence
3) Whether the sentence contains the title word
4) Sentence keyword scoring.

Finally we can get the importance score of each sentence by weighted sum above feature scores.

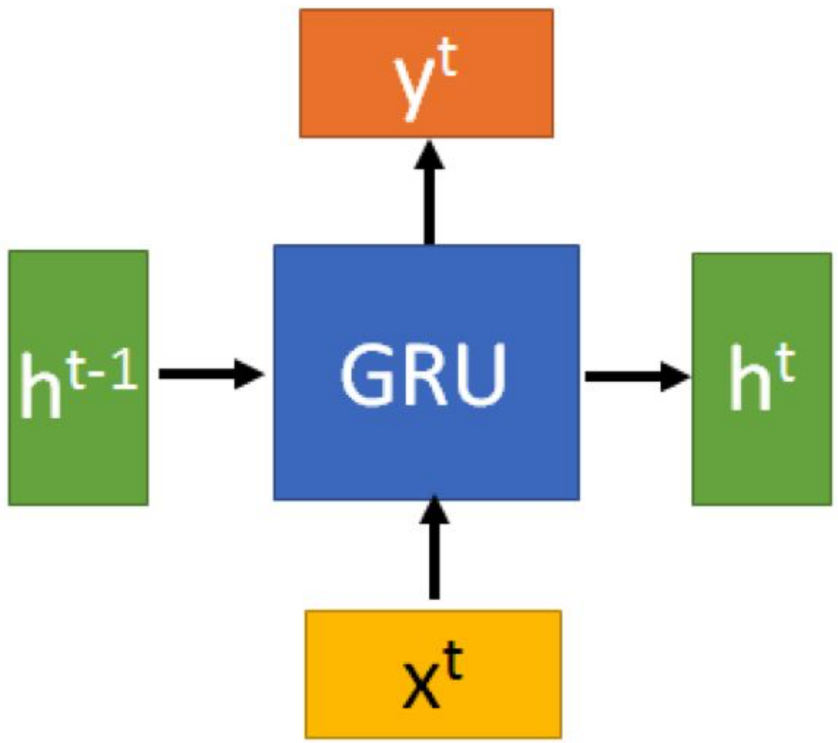
4. Output
• The output result is normally the first N sentences after sorting.

• GRU simplified introduction

The structure of GRU input and output.

The input and output structure of GRU is same as the normal RNN.
There is a current input xt, and a hidden state ht-1 passed from the previous node. This hidden state contains the information in previous step.

Combined xt and ht-1, GRU will get the output of current state output yt and the hidden state ht in current step and also can passed to the next node.

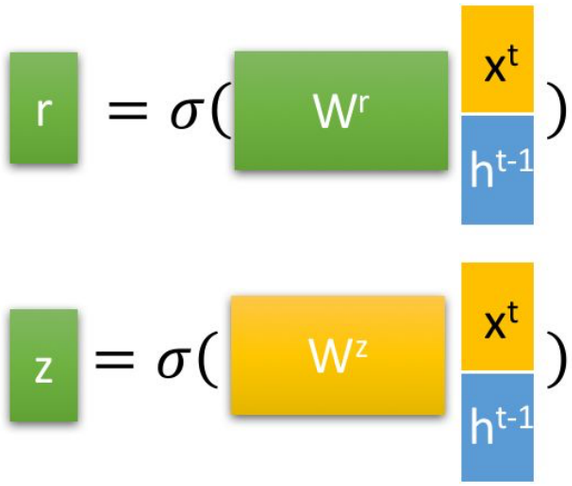


• GRU simplified introduction

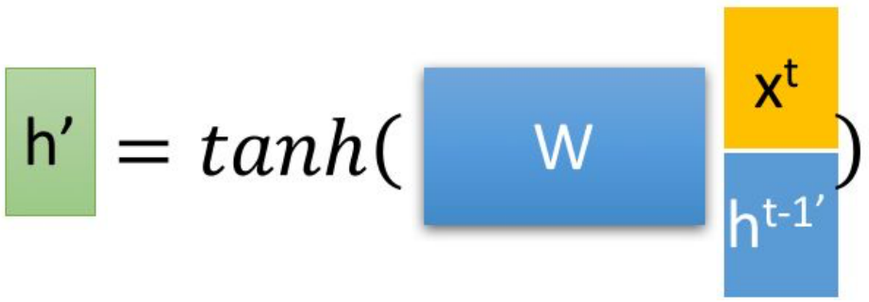
GRU internal structure

Firstly, we get the two gated states by the hidden state h^{t-1} from previous node and the input x^t of the current node. As shown in Figure below, the reset gate r is to control the reset gate, while the update gate z is to control the update gate.

Tips: σ is sigmoid function, which used to convert the data to a value in the range of 0-1 and act as a gate signal.



After obtaining the gate signal, the reset gate is used to get the 'reset' data $h^{t-1'} = h^{t-1} \odot r$, and concatenate the $h^{t-1'}$ and xt, then a tanh activation function is used to scale the data to the range of (-1, 1). That is the value h', as shown in Figure below.



The h' here mainly contains the information of current input x^t. Adding h' to the current hidden state, which is equivalent to "memorize the information of current state." This is quite similar to the memory selection phase of LSTM.

• GRU simplified introduction

“Memory update" phase.

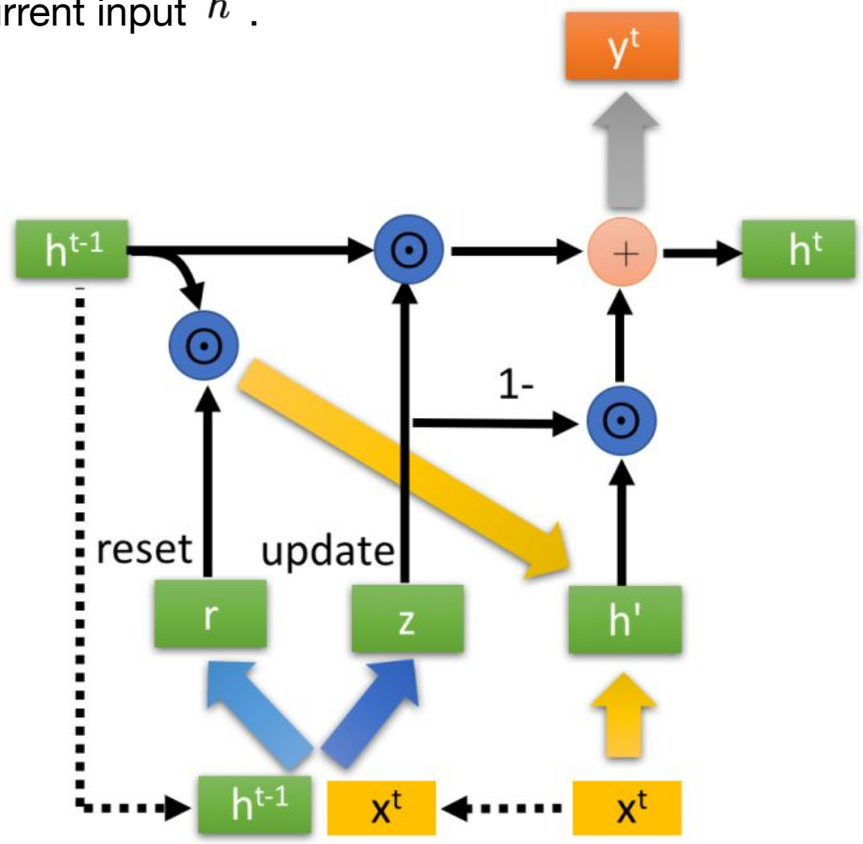
At this stage, GRU carried out two steps memory ‘forget’ and ‘memorize’ process at same time. And We use the previously obtained update gate z.

The Update formula: $h^t = z \odot h^{t-1} + (1 - z) \odot h'$

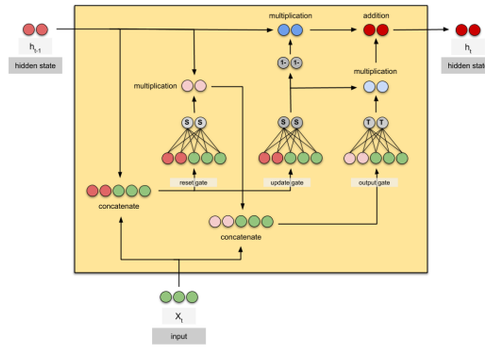
$z \odot h^{t-1}$: Indicates the selective "forget" of the original hidden state. Here you can imagine it as a forget gate, forget some unimportant information in h^{t-1}.

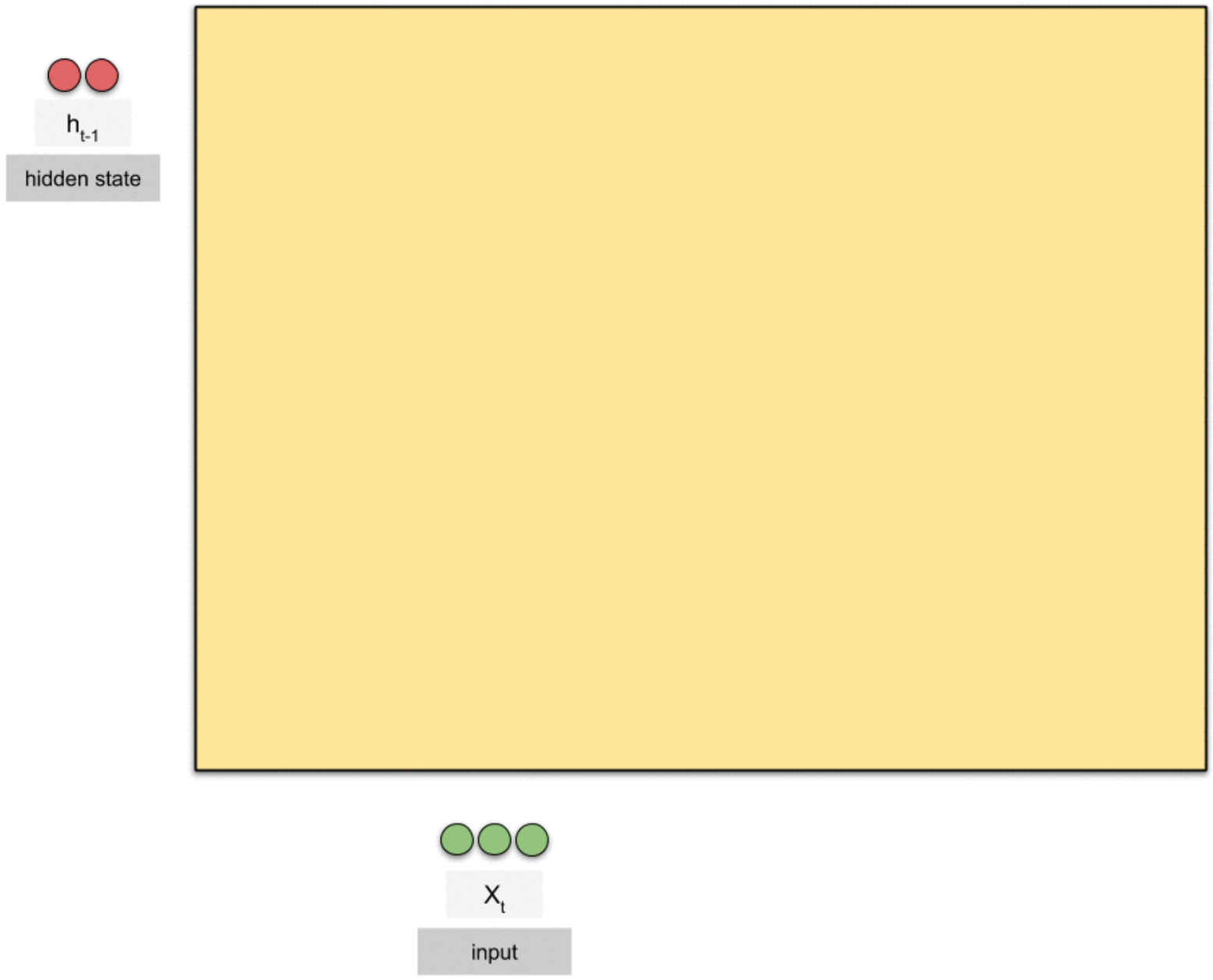
$(1 - z) \odot h'$: Indicates selective "memory" of the information in current node.

$h^t = z \odot h^{t-1} + (1 - z) \odot h'$: this step is to forget some unimportant information in previous h^{t-1}, and add some important information from the current input h'.



• GRU simplified introduction

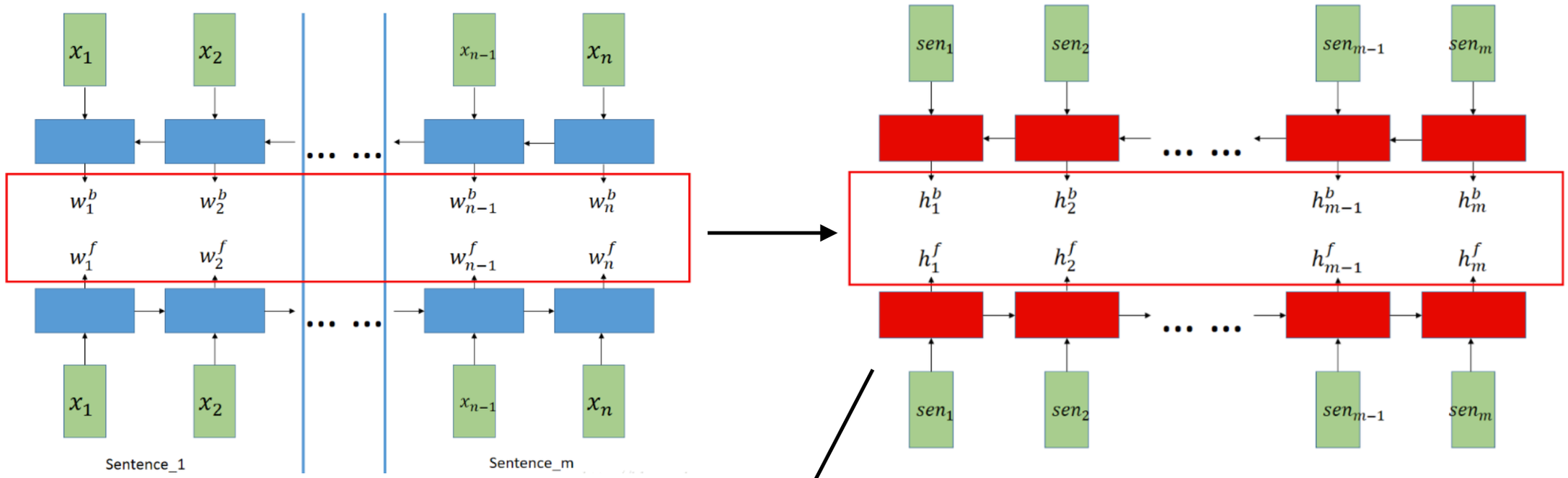




• SummaRuNNer

The author used GRU as the features extractor (representation method) of the sequence classifier.

And the model consists of a two-layer bi-directional GRU.



Bi-directional GRU in word-level

Bi-directional GRU in sentence-level

The representation of the entire document is then modeled as a non-linear transformation of the average pooling of the concatenated hidden states of the bi-directional sentence-level GRU, as shown below, formula (5).

$$\mathbf{d} = \tanh(W_d \frac{1}{N_d} \sum_{j=1}^{N^d} [\mathbf{h}_j^f, \mathbf{h}_j^b] + \mathbf{b}), \qquad (5)$$

• SummaRuNNer

Figure 1: SummaRuNNer: A two-layer GRU based sequence classifier:

The bottom layer operates at word level within each sentence, while the top layer runs over sentences. Double-pointed arrows indicate a bi-directional RNN. The top layer with 1's and 0's is the sigmoid activation based classification layer that decides whether or not each sentence belongs to the summary.

- SummaRuNNer

Logistic layer for classification

$$
\begin{aligned}
P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(&W_c \mathbf{h}_j && \text{\#(content)} \\
&+ \mathbf{h}_j^T W_s \mathbf{d} && \text{\#(salience)} \\
&- \mathbf{h}_j^T W_r \tanh(\mathbf{s_j}) && \text{\#(novelty)} \\
&+ W_{ap} \mathbf{p}_j^a && \text{\#(abs. pos. imp.)} \\
&+ W_{rp} \mathbf{p}_j^r && \text{\#(rel. pos. imp.)} \\
&+ b), && \text{\#(bias term)} \qquad (6)
\end{aligned}
$$

In Eqn. (6), the term $W_c h_j$ represents the information content of the j^{th} sentence, $h^T_j W_s d$ denotes the salience of the sentence with respect to the document, $h^T_j W_r \tanh(s_j)$ captures the redundancy of the sentence with respect to the current state of the summary, while the next two terms model the notion of the importance of the absolute and relative position of the sentence with respect to the document.

- SummaRuNNer

sj is the dynamic representation of the summary at the jth sentence position, given by:

$$\mathbf{s}_j = \sum_{i=1}^{j-1} \mathbf{h}_i P(y_i = 1 | \mathbf{h}_i, \mathbf{s}_i, \mathbf{d}). \qquad (7)$$

In other words, the summary representation is simply a running weighted summation of all the sentence-level hidden states visited till sentence j, where the weights are given by their respective probabilities of summary membership.

• SummaRuNNer

Loss function: We minimize the negative log-likelihood of the observed labels at training time.

$$
\begin{aligned}
l(\mathbf{W}, \mathbf{b}) &= -\sum_{d=1}^{N}\sum_{j=1}^{N_d}(y_j^d \log P(y_j^d = 1 | \mathbf{h}_j^d, \mathbf{s}_j^d, \mathbf{d}_d) \\
&+ (1 - y_j^d) \log(1 - P(y_j^d = 1 | \mathbf{h}_j^d, \mathbf{s}_j^d, \mathbf{d}_d))
\end{aligned} \tag{8}
$$

• Training process

Extractive Training

Problem of extractive training.

Lack of ground truth in the form of sentence-level binary labels for each document, representing their membership in the summary.

Solution:

A greedy approach that can maximize the Rouge score.

They add one sentence at one time incrementally to the summary, such that the Rouge score of the current set of selected sentences is maximized with respect to the entire gold summary.

They stop when none of the remaining candidate sentences improves the Rouge score upon addition to the current summary set. Finally return this subset of sentences as the extractive ground-truth.

• Training process

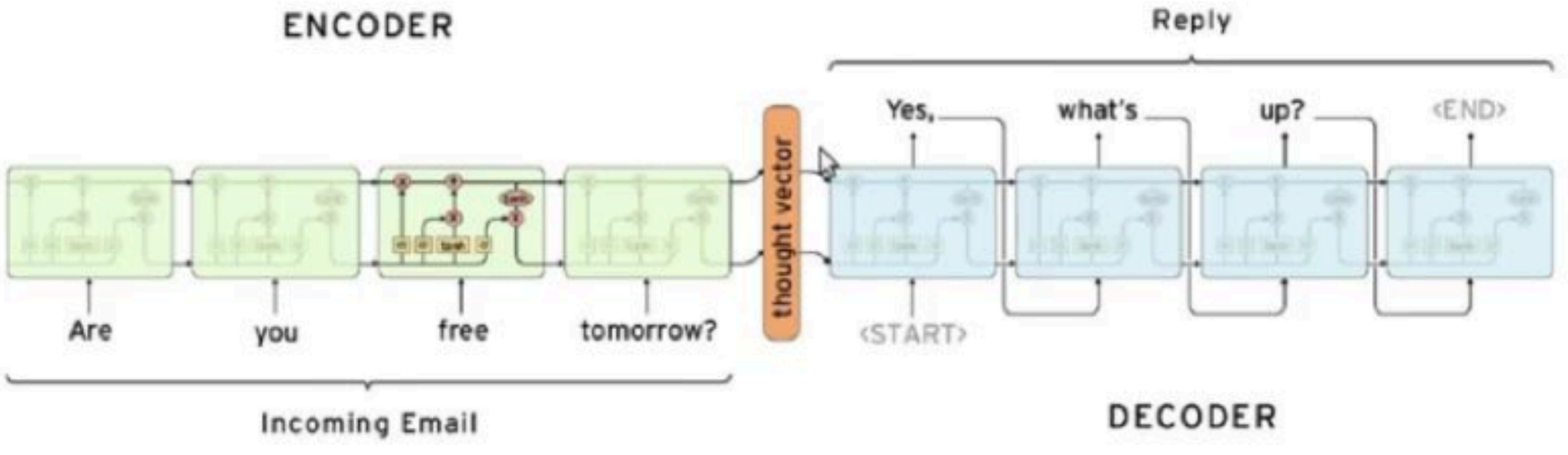
Abstractive Training

Why?
It can eliminate the need to generate approximate extractive labels.

To train SummaRuNNer using reference summaries, we couple it with an RNN decoder that models the generation of abstractive summaries at training time only. The RNN decoder uses the summary representation at the last time-step of SummaRuNNer as context, which modifies Eqs. 1 through 3 as follows:

$$
\begin{aligned}
\mathbf{u}_k &= \sigma(\mathbf{W'}_{ux}\mathbf{x}_k + \mathbf{W'}_{uh}\mathbf{h}_{k-1} + \mathbf{W'}_{uc}\mathbf{s}_{-1} + \mathbf{b'}_u) \\
\mathbf{r}_k &= \sigma(\mathbf{W'}_{rx}\mathbf{x}_k + \mathbf{W'}_{rh}\mathbf{h}_{k-1} + \mathbf{W'}_{rc}\mathbf{s}_{-1} + \mathbf{b'}_r) \\
\mathbf{h}'_k &= \tanh(\mathbf{W'}_{hx}\mathbf{x}_k + \mathbf{W'}_{hh}(\mathbf{r}_k \odot \mathbf{h}_{k-1}) + \\
&\quad \mathbf{W'}_{hc}\mathbf{s}_{-1} + \mathbf{b'}_h)
\end{aligned}
$$

where s_{-1} is the summary representation as computed at the last sentence of the sentence-level bidirectional GRU of SummaRuNNer as shown in Eq. 7.



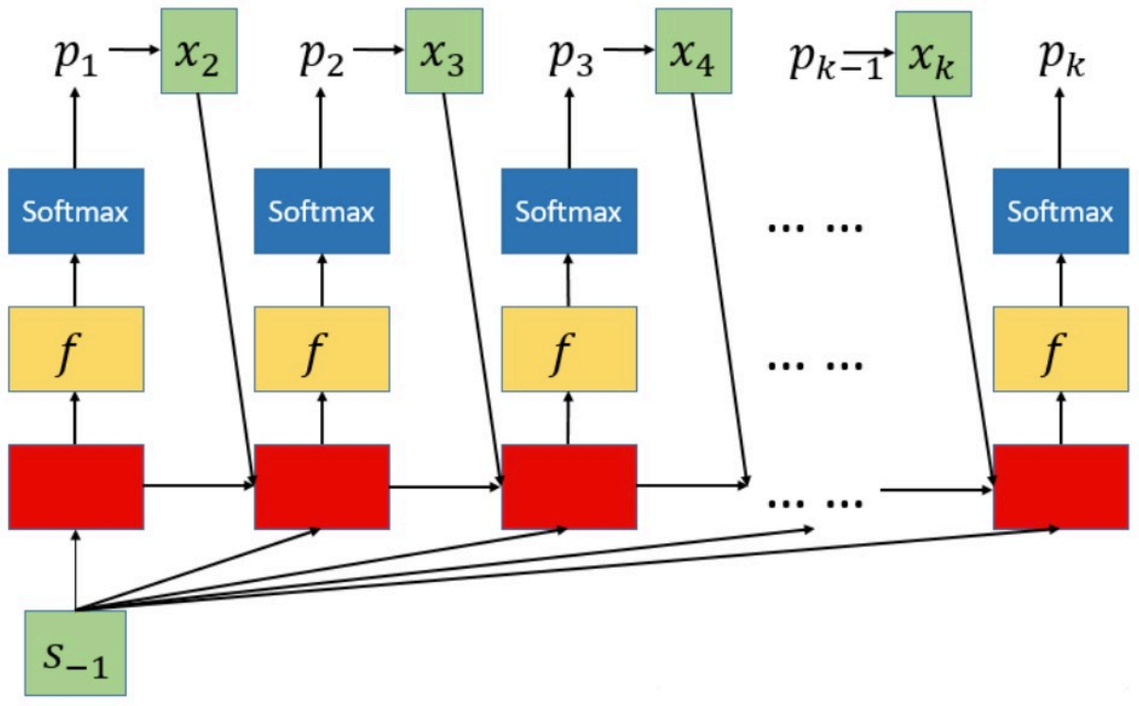
• Training process

Abstractive Training

The decoder is equipped with a soft-max layer to emit a word at each time-step. The emission at each time-step is determined by a feed-forward layer f followed by a softmax layer that assigns p_k, probabilities over the entire vocabulary at each time-step, as shown below.

$$\mathbf{f}_k = \tanh(\mathbf{W'}_{fh}\mathbf{h}_k + \mathbf{W'}_{fx}\mathbf{x}_k + \mathbf{W'}_{fc}\mathbf{s}_{-1} + \mathbf{b'}_f)$$

$$\mathbf{P_v}(\mathbf{w})_k = \text{softmax}(\mathbf{W'}_v\mathbf{f}_k + \mathbf{b'}_v)$$



• Training process

Abstractive Training

Loss function:
Instead of optimizing the log-likelihood of the extractive ground truth as shown in Eq. 8, we minimize the negative log-likelihood of the words in the reference summary as follows.

$$l(\mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}') = -\sum_{k=1}^{N_s} \log(\mathbf{P_v}(w_k))$$

where $\mathbf{N_S}$ is the number of words in the reference summary.

How the abstractive training helps for the extractive sammarization?

Since the summary representation s−1 acts as the only information channel between the SummaRuNNer encoder model and the decoder, maximizing the probability of abstractive summary words as computed by the decoder will require the model to learn a good summary representation **s-1** which in turn depends on accurate estimates of extractive probabilities p(yj) at same time.

• Experiment & Setting & Evaluation

Corpora

- CNN/DailyMail： 196557 training documents, 12147 validation documents and 10396 test documents. On average, about 28 sentences per document in the training set, and an average of 3-4 sentences in the reference summaries.

- DUC2002：567 documents；

Evaluation

- Rouge metric: Rouge-1、Rouge-2、Rouge-L

SummaRuNNer Settings

- 100-dimensional word2vec embeddings trained on the CNN/Daily Mail corpus as initial embedding.
- Limited the vocabulary size to 150K and the maximum number of sentences per document to 100, and the maximum sentence length to 50 words, to speed up computation.
- Fixed the model hidden state size at 200. Batch size is 64 at training time, and adadelta to train our model.
- Trained SummaRuNNer both extractively as well as abstractively.

• Experiment & Setting & Evaluation

Gold Summary: Redpath has ended his eight-year association with Sale Sharks. Redpath spent five years as a player and three as a coach at sale. He has thanked the owners, coaches and players for their support.	Salience	Content	Novelty	Position	Prob.
Bryan Redpath has left his coaching role at Sale Sharks with immediate effect.	0.1	0.1	0.9	0.1	0.3
The 43 - year - old Scot ends an eight-year association with the Aviva Premiership side, having spent five years with them as a player and three as a coach.	0.9	0.6	0.9	0.9	0.7
Redpath returned to Sale in June 2012 as director of rugby after starting a coaching career at Gloucester and progressing to the top job at Kingsholm .	0.8	0.5	0.5	0.9	0.6
Redpath spent five years with Sale Sharks as a player and a further three as a coach but with Sale Sharks struggling four months into Redpath's tenure, he was removed from the director of rugby role at the Salford-based side and has since been operating as head coach .	0.8	0.9	0.7	0.8	**0.9**
'I would like to thank the owners, coaches, players and staff for all their help and support since I returned to the club in 2012.	0.4	0.1	0.1	0.7	0.2
Also to the supporters who have been great with me both as a player and as a coach,' Redpath said.	0.6	0.0	0.2	0.3	0.2

Figure 2: Visualization of SummaRuNNer output on a representative document. Each row is a sentence in the document, while the shading-color intensity is proportional to its probability of being in the summary, as estimated by the RNN-based sequence classifier.

In the columns are the normalized scores from each of the abstract features in Eqn. (6) as well as the final prediction probability (last column).

Sentence 2 is estimated to be the most salient, while the longest one, sentence 4, is considered the most content-rich, and not surprisingly, the first sentence the most novel. The third sentence gets the best position based score.

• Experiment & Setting & Evaluation

	Rouge-1	Rouge-2	Rouge-L
Lead-3	21.9	7.2	11.6
LReg(500)	18.5	6.9	10.2
Cheng *et al* '16	22.7	8.5	12.5
SummaRuNNer-abs	23.8	9.6	13.3
SummaRuNNer	**26.2**±0.4*	**10.8**±0.3*	**14.4**±0.3*

Table 1: Performance of various models on the entire Daily Mail test set summary length restricted to 75 bytes.

	Rouge-1	Rouge-2	Rouge-L
Lead-3	40.5	14.9	32.6
Cheng *et al* '16	**42.2**	**17.3**	**34.8***
SummaRuNNer-abs	40.4	15.5	32.0
SummaRuNNer	42.0 ±0.2	16.9 ±0.4	34.1 ±0.3

Table 2: Performance of various models on the entire Daily Mail test set with summary length restricted to 275 bytes.

Cheng et al,16 designed a rule- based system that determines whether a document sentence matches a highlight and should be la- beled with 1 (must be in the summary), and 0 otherwise manually.

• Conclusion

Most important contribution of this paper.

- Proposed a RNN based sequence classifier: SummaRuNNer, which is quite simple and interpretable.
- Proposed a novelty abstractive training mechanism to eliminate the need for extractive labels at training time.

Further work.

Author plans to further explore combining extractive and abstractive approaches as part of future work.

One simple approach could be to pre-train the extractive model using abstractive training.

Further, they also plan to construct a joint extractive-abstractive model where the predictions of their extractive component form stochastic intermediate units to be consumed by the abstractive component.

• Problems

- If we work on sentence level, how sensitive is the model when it comes to sentences splittings errors, as sentence splitting is not a trivial task?

  Normally we splitting sentence by some specified punctuation, like full stop mark, question mark, etc. Even there is sentence splitting error, I think it will not have much influence on the extractive summarization,

- Does the greedy approach described in 2.1 has a beam search setup, as used in nmt?

  No, they only used greedy method, instead of beam search and no such setup in greedy progress.

- How are the scores in Figure 2 computed exactly? More specifically, what are they normalized against? And where did the other position score go? (I guess, they added the position scores (before normalizing probably), but I don't see where they say that.)

- Taking the intuitive names content, saliency and novelty atface-value, I would expect long sentences to get high scores, because they tend to include lots of new information. But from my own intuition,it seems like they would also tend to include lots of uninteresting information. How is this accounted for? Maybe parts of the content,saliency and novelty scores also include something like length or tend to penalize too much information? Then again, the long sentence in Figure 2 did get a very high score.

- I am wondering why GRU instead of LSTM where used in the "SummaRuNNer" paper, especillay as in "Neural Summarization by Extracting Sentences and Words" they compare their work to uses LSTMs. Is this maybe because of bidirectionality in their architecture? But there are also Bidirectional LSTMs.

  More simple and efficient for training but get comparable result of LSTM.

• Problems

Further discussion

- I think that calling the rules to select which sentences to extract in “Neural Summarization by Extracting Sentences and Words” handcrafted features is wrong. In SummaRuNNer the authors use ROUGE scores instead of unigram and bigram overlap between sentences and highlights. Both methods do not seam like handcrafted features to me.

- The authors from this paper claim, that their model is very interpretable. According to this, one expect an examination of features, which is not given. In my opinion, future work of SummaRuNNer should also concentrate on this study.

- For the Nallapati paper I'm wondering how much sense it makes to have the summarization vector be all the sentence vectors scaled by their probability of being included. For the method where they train an abstractive model on top of their extractive model it makes sense so that they can backpropagate through the decision to include a sentence or not, but with the extractive model, since it has to make a hard decision anyways, it seems like the sentence vectors shouldn't be scaled at all before being added to the summarization vector.