# Sentient: Abstractive Text Summarization using Sequence-to-Sequence RNNs

**Anitha Ranganathan, Saugat Chetry, Sweta Thapliyal**
**Department of Computer and Information Sciences, University of Florida – Gainesville, FL**

## Introduction

**Text Summarization is a process of summarizing a large text file by taking input and producing a precis of that file. The output of summarization is a semantically correct text which maintains the original essence of the document. As of now the extractive text summarization is prevalent in industry which is like highlighting important words in the document. There are no out of vocabulary words in the output. These types of model fail to handle the semantics of the document and the key ideas in the document. Abstractive text summarization, on the other hand, is quite complex but very effective. We have tried to achieve this in our project for news summarization. The main summarizer is implemented using deep learning and recurrent neural network. At the core of this news summarizer will be attentional encoder decoder neural network. Since we are using an abstractive model the for summarization, our outputs capture the pith of the input text. Due to the inclusion of pointer networks we are able to handle the out of vocabulary words as well and with the help of coverage mechanism the summary generated is free of repetitive statements. We have provided web application which takes the news, summarize it and display it to the user and on click delivers the full article as well.**

## Problem statements and current systems

### 1. Drawbacks of extractive summarization:

Text summarization systems are mostly focused on extractive approach. The draw backs of this approach basically motivated us to build an abstractive one. Drawbacks of most of the summarization systems are

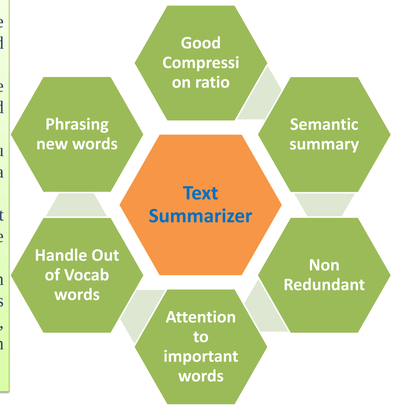
1.Simply highlighting
2.No phrasing of complex and long sentences
3.Not semantically related
4.Poor compression ratio
5.Difficulties in handling out of vocab words
6.Redundant summary

### 2. Motivation for sentient summarizer:

All these issues are the problem basis of the problem statement of our text summarizer.

**RELATED WORKS:**

1. Nallapati et al[1] provided the first baseline model and the CNN dataset using encoder and decoder for sequence to sequence models.
2. Pointer networks[5] combine the extractive and abstractive used for the rare words and OOV word problems in machine Translation.
3. Sequence to sequence models uses bhadanau et al[2] soft attention strategy to generate a well balanced summary.
4. The coverage mechanism was used by xu et al[3] for video captioning to reduce the redundancy.
5. Attention mechanism can be applied in various techniques. There have approaches like NMT and summarization (Nallapati et al., 2016) who have explored temporal attention as an alternative.




Sentient is a text summarization system that uses all of the above approaches combined in a way that facilitates the text summarization. We used the base line model and added the coverage mechanism and the pointer networks to generate the semantic text summary.

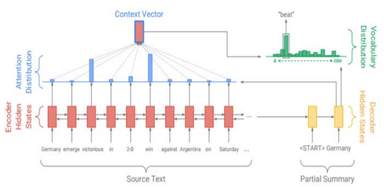



Figure explains baseline Sequence to sequence model with soft attention. Model generates OOV words too. For eg. It scans the iput sequence 'Germany emerge victorious in 2-0 win against Argentina on Sunday' and generates the word 'beat' as part of its abstractive summary by using attention on the word 'victorious'.[4]

## Sentient Summarizer

### 1. Baseline sequence to sequence unit:

1. Sequence to sequence models are the models which takes an input sequence and returns an output sequence.
2. In our project, the input sequence refers to the original text and output sequence is the summary generated.
3. We can use any RNN for this purpose, however, we choose to feed the article tokens wi, into a single bidirectional LSTM which acts as our encoder. Like all other feed forward networks, the encoder provides an output encoder hidden state hi. The attention distribution is as below where v, Wh, Ws and battn are learnable parameters.

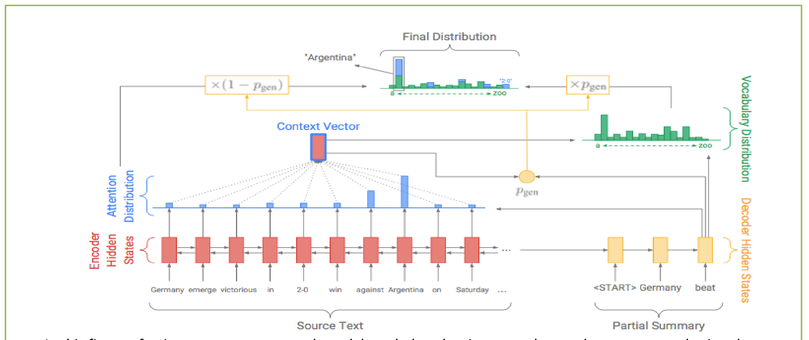
$$e^t_i = v^T \tanh(W_h h_i + W_s s_t + b_{attn}) \quad (1)$$
$$a^t = \text{softmax}(e^t) \quad (2)$$

4. The attention vector is given as

$$h_t^* = \sum_i a^t_i h_i \quad (3)$$

5. We have modified the concatenated pairs in the Bahdanau et al[6] to:

$$P_{vocab} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b') \quad (4)$$




*In this figure of pointer generator network model, each decoder time-step the words are generated using the probability generator function which calculates using the below equations .[4]*

### 2. Pointer Generator unit:

1. We use a pointer generator unit which is well known for its hybrid functionality between extractive and abstractive summarization capabilities.
2. At every time-step at the decoder we use the sigmoid distribution on the context vector, decoded state and the decoder input to obtain a probability generation pgen. As a result of this we allow copy and vocabulary distribution from a fixed vocabulary.

$$p_{gen} = \sigma(w^T_h * h_t^* + w^T_s s_t + w^T_x x_t + b_{ptr}) \quad (5)$$

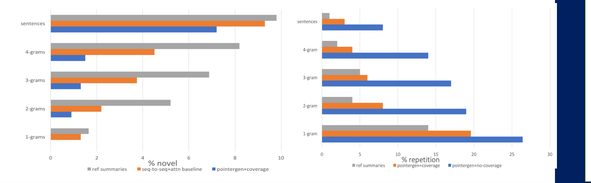
### 3. Coverage mechanism:

1. To solve the problem of repetition which is present in the sequence-to-sequence model, a coverage vector is maintained.
2. The coverage vector represents the degree of coverage the words in the source document have received so far from the attention mechanism.
3. This coverage vector is fed to the attention mechanism as an additional input which ensures that the current decision of the attention mechanism is based on its previous decision which will in-turn help generating repetitive text. To penalize for repeatedly attending the same location, a coverage loss is defined as:

$$\text{covloss}_t = \sum_i \min(a^t_i, c^t_i) \quad (6)$$

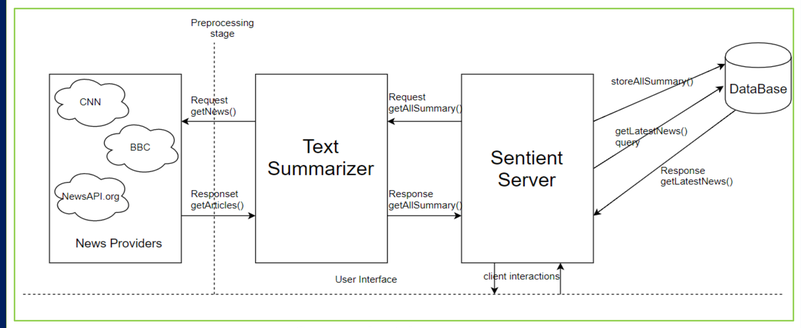
## Observation

Setup:
256 dimensional hidden layer
128 dimension word embeddings
Training on 50k words
No pretraining for word embeddings
Learned from scratch
Length of summary – 100 tokens
Length of articles 400 words




## Implementation


**Basic Flow of the system**

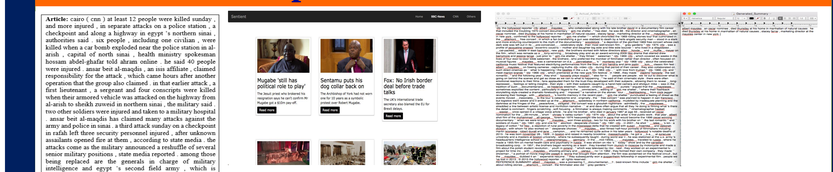



**System Architecture**

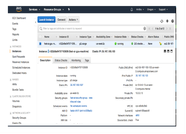
**SYSTEM COMPONENTS:**

1. **Sentient preprocessor: Preprocess the articles in tokenized binary format.**
2. **Sentient text summarizer: converts the binary encoded articles to summaries which maintains the essence of the article.**
3. **Sentient news server: mainly responsible for providing the summarized news to the client of our application.**
4. **Sentient smart storage: will be saving all the summarized news.**

## Experiment and Results




| | ROUGE 1 | ROUGE 2 | ROUGE L |
|---|---|---|---|
| Abstractive Model | 35.4 | 13.1 | 33.1 |
| Base line model | 31.3 | 11.7 | 28.0 |
| Pointer-generator | 36.2 | 15.5 | 32.1 |
| Pointer-generator + coverage | 39.1 | 17.1 | 36.2 |



## References

[1] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Association for the Advancement of Artificial Intelligence*.
[2] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2015. End-to-end attentionbased large vocabulary speech recognition. CoRR, abs/1508.04395.
[3] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International Conference on Machine Learning*. 2015
[4] Abigail See, Peter J. Liu ,Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks
*[5] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.*
[6] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).