Document Summarizer using Skip-Thought and Content Based Clustering

Dr. Siddaraju[1], Manu Hegde[2], Harshal Bhatia[3], Kalpitha V Bekal[4], Mohan Krishna S[5]
Department of Computer Science and Engineering, Dr.Ambedkar Institute of Technology, Bengaluru-56
[1]siddaraju@dr-ait.org, [2]hegde.manu@dr-ait.org, [3]bhatia.harshal@dr-ait.org, [4]kalpithabekal@dr-ait.org, [5]krishna.mohan@dr-ait.org

***Abstract*— An approach that can help resolve issues faced by many known models i.e loss of context over a large document, repetition and coverage tracking by using output vectors from GRU based encoder known as Skip-Thought encoder to map sentences to higher dimension based on their content similarity and then cluster them. These clusters are then fed separately to a Seq2Seq encoder-decoder to yield summary. This approach aims for a model that can operate with large documents without a quadratic increase in computational resource requirements.**

***Keywords*— Sequence2Sequence, Word2Vec, Glove, BERT, GPT, UNK, Skip-Thought, T-SNE, K-Means, RNN, LSTM, GRU, Encoder, Decoder.**

I. Introduction

Document summarization has always been an issue on which a large number of techniques have been applied and tested. Until 2013, all major techniques for document summarization were exclusively extractive with specific hand engineered feature detectors . They usually encompassed a complex data pipeline that included processing techniques specific to a given language [1].

With the recent resurgence of neural networks and the creation of high dimensional word embeddings like Word2Vec[2] and Glove[3] enabled neural networks to outperform traditional techniques with minimal supervision and without any manual feature engineering. This enabled the creation of language agnostic systems that learn to represent information appropriately for a given language they are trained on[4].

II. Related Work

All major neural network based architectures depend upon one-hot encodings of embeddings like Word2Vec or GloVe. However, whenever there is a new word, especially in case of nouns, the typical Sequence2Sequence encoder decoder models output UNKs i.e unknowns. Various efforts have been made to enable copy mechanisms like pointer generator networks [5]. Pointer generator network also tries to address the problem of repetition seen in typical encoder decoder models.

Transformer models like BERT[6] and GPT[7] have succeeded at this task but at the cost of very expensive computation and many hundreds of days of gpu training. Subsequently, such models are memory and resource intensive.

Skipthought encoder[8], is a GRU based encoder that generates fixed length learned representations for every input sentence. It generates a 4800 dimensional embedding vector for each sentence. This enables one to hence use these vectors as reliable and powerful representations of the contents of any given sentence itself rather than having to process a very large embedding or one hot encoded vector for every single token/word. Also it encompasses a vocabulary expansion technique for previously unknown words. This means it can give meaningful output even when it encounters new words. Also, unlike many other models, this does not grow the model size quadratically with vocabulary size.

III. Drawbacks of Existing Models

The existing models are of following types.

- *Purely extractive rule based systems:* These systems need to be re-designed for every language and features to be used have to be decided manually, thereby failing to capture the overall meaning and context in larger documents.

- *Purely Recurrent Encoder Decoder Networks:* Encoder Decoder networks based on variants of RNN like LSTM and GRU. These networks have the main drawback of repetition and coverage. This means that the summary yielded contains repeated phrases, sentences and either omit or repeat topics, contexts

present in the source document. Another drawback is the inability to handle new words, i.e words previously not seen during training resulting in outputting 'UNK' in place of such words. This results in summaries that are incomplete and unable to convey the full meaning of the document. But even more importantly, the information from these words is lost, leading to potential misunderstanding and mis representation of information.

- *Models with explicit copy mechanism:* Models like Pointer Generator Networks have explicit, probabilistic copy mechanisms where the model tries to copy a word from source text if the decoder is unable to produce output for a portion of text that contains words not seen by the model during training. These models very often end up copying text from source document even when there are no unknown words. They also need explicit mechanism to track coverage of source document to prevent repetition.

- *Transformers and Other Hybrid Models:* These models try to solve the problem of copying, but do not perform well in case of context coverage. They often tend to mix up sentences of different contexts. Although their output is semantically correct, they can end up generating misinterpreted summaries. They are also one of the most computationally intensive models employed for summarization.

IV. Proposed Model

The proposed model operates as follows.

- Obtain embeddings for every sentence.
- Reduce dimensionality of embeddings using T-SNE.
- Cluster the embeddings using K-Means.
- Obtain one summary for each cluster separately.

A. Obtain Embeddings for every Sentence

Skip-Thought encoder to obtain fixed length sentence embeddings, for sentences of varying length. This can step is necessary to obtain meaningful representation of a sentence in form of a vector i.e an embedding in a higher dimension, such that the sentences with similar meaning are mapped close to each other and vice versa. Also this space is such that a dot product between two sentences can be used to obtain a measure of their similarity or dissimilarity. The length of these vectors is 4800.

Using Skipthought sentence embeddings can solve the misunderstanding of sentences due to UNKs, because of its vocabulary expansion capability to handle previously unknown words.

B. Reduce Dimensionality of Embeddings using T-SNE

T-SNE (t-Distributed Stochastic Neighbour Embedding)[9] is widely known algorithm for dimensionality reduction. The T-SNE algorithm is designed to project a set of points in higher dimension to a lower dimension, yet trying to minimize the loss in pairwise distances between all set of input points.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)},$$

Where $p_{j|i}$ is the probability density under a Gaussian centered at x_i.

Since the sentence embeddings each are of length 4800. Effectively projecting them to lower dimension without loss of information is necessary and would lead to better performance. This would result in much better result from K-Means Clustering. Hence T-SNE was chosen over other algorithms like SVD or PCA.

C. Cluster the embeddings using K-Means

K Means is used for clustering the sentence embeddings. The number of clusters is directly related to the desired length of summary.

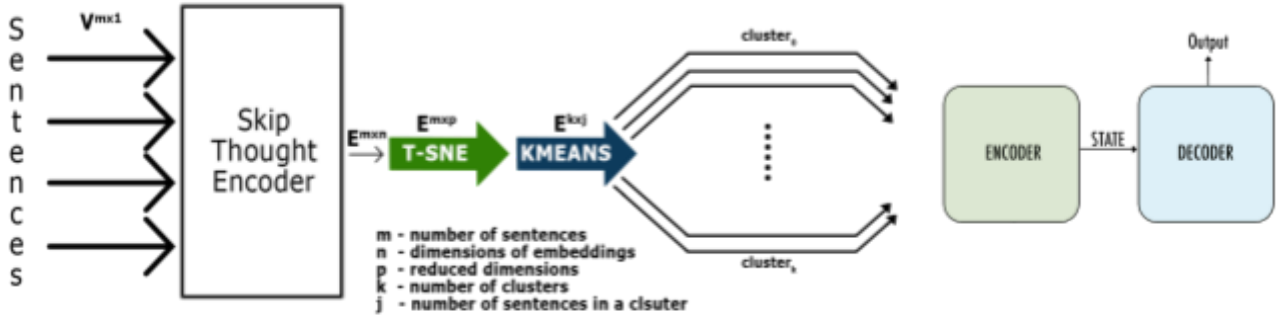
Usually this number is chosen as square root of the number of sentences in the document. This can solve the problem of loss of context as well as requirement of large computational resource for large documents.

Here, each cluster is formed such that all sentences in a cluster have similar meaning or are very closely related to each other. Hence the document is effectively divided based on content similarity.

D. Obtain one summary for each cluster separately

Finally, set of sentences in each cluster are fed to an encoder decoder model. This helps in solving the problem of

Fig 1. Proposed Model Architecture



coverage tracking. Also since all sentences in a cluster have similar meaning or are closely related to each other, the encoder decoder model would process sentences of same or similar topic. This would allow for a model to be used that is computationally less intensive due to smaller input size and smaller context. There are different variants of attention. Two main variants are Additive and Multiplicative attention.
The following equation describes Multiplicative attention.

$$e_i = v^T tanh(W_1 h_i + W_2 s) \epsilon R$$

Where e_i is the attention vector, W_1, W_2 are weight matrices and v is a weight vector.

Segregation of context or topic is of great significance since assumptions can be made in case of unknown words.
An encoder decoder model consisting of Bidirectional LSTM and attention decoder[10] would generate the final summary for each cluster independently. Additionally Beam search could be incorporated for better results as opposed to Greedy decoding that is in use currently.

V. Novelty in Proposed Model

The main idea of the proposed model is to logically divide up the problem space and use suitable, specialized and exclusive techniques, instead of deploying one single encoder decoder model that is responsible for the entire task resulting in poor performance.

$$\begin{aligned}
&Algorithm\ Summarize\\
&Input : Document\\
&Output : Summary\\
&\\
&S < -summarize_document(D)\\
&\quad p < -number\ of\ sentences\ in\ document\\
&\quad q < -output\ size\ of\ skipthought\ encoder\\
&\quad r < -reduced\ dimensions\ after\ TSNE\\
&\quad E_{pXq} < -Array(p, q)\\
&where\ E_{pXq}\ is\ Embeddings\ Matrix\\
&\quad Summary < -Array()\\
&\quad for\ s_i\ \epsilon\ document\\
&\qquad e_i < -skipthoughts(s_i)\\
&\qquad E_{pXq}.append(e_i)\\
&\quad E_{pXr} < -tsne(E_{pXq}, dimensions = r,\\
&\quad perplexity = 30, iters = 1000)\\
&\\
&num_of_clusters < -cluster_strategy(p, E_{pXr})\\
&clusters < -KMeans(E_{pXr}, num_of_clusters)\\
&for\ c_i \epsilon\ clusters\\
&\qquad summ_i < -summarize_cluster(c_i)\\
&\qquad Summary.append(summ_i)\\
&return\ Summary\\
&\\
&cluster_strategy(p, E_{pXr})\\
&\qquad strategy < -square_root\\
&\qquad if strategy = square_root\\
&\qquad\quad return\ sqrt(p)\\
&\qquad else\\
&\qquad\quad return\ elbow_method(E_{pXr})\\
&\\
&summarize_cluster(cluster)\\
&\quad mini_doc < -str_concat(cluster)\\
&\quad hidden_states < -encoder(mini_doc)\\
&\quad decoder_output < -attention_decoder(\\
&\quad hidden_states)\\
&\quad Summary < -Array()\\
&\quad for\ h_j\ \epsilon\ hidden_states\\
&\quad summ_i < -greedy_decoder(h_j)\\
&\quad Summary.append(summ_i)\\
&\\
&Where\ elbow_method\ is\ a\ technique\ to\ find\\
&ideal\ number\ of\ clusters\ in\ the\ data
\end{aligned}$$

- It solves the problem of coverage by separating i.e clustering sentences based on its meaning.
- Clustering also solves the problem of repetition. Simply because the attention based encoder decoder network will be mandated to produce one single summary for a given topic.
- It tries to solve problem of UNKs causing mis representation of information by using a vocabulary expandible encoder.
- It requires much lesser computational resources compared to other models.

VI. Implementation

The model used skip-thought embeddings of size 4800 for a sentence. This is due to the use of two unidirectional GRUs, each yielding a 2400 size vector. Hence one vector will contain embeddings for forward direction and the other for backward direction. Hence making it a Bi-Directional network.

This model is proposed with more focus on making a model that is deployable and scalable. Hence, resource

requirements do not directly depend on the size of the input document. However the size of the vocabulary has a direct effect on the size occupied by the model in memory. Hence the model was tested with vocabulary size 50,000. Skip-thought encoder as well as the encoder-decoder with attention was trained on amazon food reviews dataset.

It was re concurred that feeding reversed sentences helped improve the output. For the attention decoder there can be different variants for implementing attention mechanism. This model uses the variant that includes tanh nonlinearity though this adds an extra set of parameters to train. It was also found that T-SNE with perplexity 30 well given the fact that even if the document is lengthy, the number of sentences and hence number of points for T-SNE is always under 1000. Since T-SNE and K-Means are not differentiable, this model cannot backpropagate loss end to end. As an alternative, Deep Adaptive[11] clustering technique can be implemented.

VII. Conclusion

The proposed architecture is built upon some of the existing models in an attempt to overcome their limitations, yet without a significant increase in resource requirements. It tries to solve the problem of loss of information from unknown words or UNKs and of repetition and coverage tracking by using K-Means clustering. This approach yield a model that though not trainable a fully differentiable model, i.e is not a continuous function, can be trained end to end considering parameters for clustering algorithms as hyper parameters.

Acknowledgement

We express our deepest gratitude to Dr. Ambedkar Institute of Technology for having provided us with the required infrastructure and a conducive environment.

References

1. *Kupiec, Julian & O. Pedersen, Jan & Chen, Francine. (1995). A Trainable Document Summarizer. SIGIR. 68-73.*

2. *Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. (2013). Efficient Estimation of Word Representations in Vector Space.*

3. *Jeffrey Pennington, Richard Socher, Christopher D. Manning. (2014). GloVe: Global Vectors for Word Representation.*

4. *Ilya Sutskever, Oriol Vinyals, Quoc V. Le. (2014). Sequence to Sequence Learning with Neural Networks.*

5. *Abigail See, Peter J. Liu, Christopher D. Manning. (2017). Get To The Point: Summarization with Pointer-Generator Networks.*

6. *Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

7. *Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei. (2019). Language Models are Unsupervised Multitask Learners.*

8. *Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, Sanja Fidler. (2015). Skip-Thought Vectors.*

9. *Laurens van der Maaten, Geoffrey Hinton. (2008). Visualizing Data using t-SNE.*

10. *Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. (2016). Neural Machine Translation by Jointly Learning to Align and Translate.*

11. *Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, Chunhong Pan. (2017). Deep Adaptive Image Clustering.*