Knowledge graph assisted extreme summarization of scientific documents in hyperbolic space

by
Asmita Mukherjee

Under the supervision of
Dr. Tanmoy Chakraborty & Dr. Md. Shad Akhtar

Submitted in partial fulfillment of the requirements for the degree of Master of Technology, CSE-AI

Department of Computer Science Engineering
Indraprastha Institute of Information Technology - Delhi
Dec, 2022

Certificate

This is to certify that the thesis titled *"Knowledge graph assisted extreme summarization of scientific documents in hyperbolic space"* being submitted by **Asmita Mukherjee** to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Dec,2022

Dr. Tanmoy Chakraborty & Dr. Md. Shad Akhtar

Department of Computer Science Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi 110 020

Acknowledgements

I am grateful to my advisor Dr. Tanmoy Chakraborty and my co-advisor Dr. Md. Shad Akhtar for guiding me throughout my thesis journey and for providing me the opportunity to do independent research project in the field of text summarization. My journey was also made a lot easier with the help of my PhD guide Yash Kumar Atri. The work would not have been made possible without the constant and steady support of my batch mates,lab mates and my family members.

Abstract

A research paper is a document that presents an original work and introduces new concepts and makes interconnections between them via arguments and statements. Extreme Summarization involves high compression of the information content of the document and representing it in a concise, meaningful format. Hence extreme summarization of scientific documents entails representation of the key concepts as presented in the concerned document in a concise and coherent manner.The challenge of this task is to capture the essential concepts as presented in the entire paper. In this paper we handle the problem of abstractive extreme summarization of scientific documents.The technique of abstractive summarization would allow us to concisely represent the information present in the entire scientific research paper by generating a new sentence rather than being constrained to having to pick sentences that is already present in the document. Since no single sentence in the document can capture the entire information presented in the document. In order to do effective extreme summarization of scientific documents,we propose Knowledge graph assisted hyperbolic BART(**KAHB**) , a knowledge graph assisted sequence to sequence architecture while transforming the intermediate embeddings into hyperbolic space. A knowledge graph helps to capture the interconnection between the concepts as presented in the paper, which a plain sequence to sequence model fails to do. Transforming the embeddings to an hyperbolic space helps to capture the inherent hierarchical relationships present in the document .Applying the above methods , we are able to achieve improvements in the performance of a standard sequence to sequence mode.

Contents

List of Figures

Chapter 1

Introduction

The main goal of this paper is to produce TL;DR i.e one line summary from research papers. In order to evaluate our methods we chose the SCITLDR dataset,which is a multi-target dataset i.e both author-written and expert-derived TLDRs of 5.4K TLDRs over 3.2K papers. We explored abstractive summarization techniques in this paper, as an extractive approach of picking a sentence or multiple sentences from the document will not be able to produce an effective summary that is able to present the key concepts demonstrated in the document. The main challenges of this task involves :

1) Processing a full text scientific document. A full text scientific document with average token length exceeding well above 5000 tokens on average, cannot be processed by the current state of the art summarization models. Most of the current state of the art models for summarization uses pretrained models which are limited to 512 or 1024 lexical tokens.Hence novel methodologies needs to be undertaken in order to capture the information presented in the full text of the document.

2) Identifying the key concepts presented in the paper. A scientific document is often a collection of concepts and interconnections between them. Concepts are often not presented explicitly but made clear via the arguments and discussions in the paper.

3) Coherence. It is not sufficient to identify the key concepts but also to represent them in a coherent and meaningful manner.

1.1 Motivation

The task of TLDR generation is of industrial and real world relevance. Since the amount of scientific discoveries along with the number of scientific paper published increases day by day, a short TLDR saves a researcher's time by allowing him/her to have a

glimpse of the key content present in the paper and thus help to make the decision of whether to delve deeper or not.

1.2 State of the art and Limitations

The current state of the art involves the use of pretrained models which limits itself to the token length of 512 and 1024. However our data has a max token length of about 10000, hence the full text information cannot be captured by them. We also hypothesize that there is a loss of hierarchical information when the embeddings are present in the 2-D dimensions.In order to fully capture the hierarchical information

1.3 Contribution

The contributions of the work done in this paper is summarized as below:

1) Incorporation of full text information in the form of knowledge graph. Creation of knowledge graph was done in a step wise format which involved co reference resolution,Sentence simplification,entity and relation extraction.

2) Transformation of intermediate embeddings from the layers of the sequence to sequence models to hyperbolic space, via poincare normalization. We hypothesize that it would enable us to capture the inherent hierarchies present in the document.

3) The SCITLDR dataset for abstract,introduction,conclusion (AIC) was augmented with the full text knowledge graph triples.

Chapter 2

Dataset

In order to evaluate our methods , we use the SCITLDR dataset which contains both author-written and expert-derived TLDRs. SCITLDR, is multi target dataset with 5,411 TLDRs spread over 3,229 academic papers in the field of computer science. Each of the 1,992 papers in the training set has a single gold TL DR. There are 619 and 618 papers in each of the development and test collections, with 1,452 and 1,967 TLDRs, respectively. The dataset is further divided in abstract only(SciTLDR-A) , abstract intro conclusion(SciTLDR-AIC) , and full text(SciTLDR-FullText) where the source document provided contains only the text from the abstract, abstract introduction and conclusion and the full text respectively. In our work we have utilized SciTLDR-FullText and SciTLDR-AIC.

2.1 Dataset Analysis

As seen in fig2.1, the source text in the AIC dataset has a maximum token count of 2000 which cannot be considered to be a short document as it lies beyond the lexical token limit of modern pretrained models. The AIC does contain the most key information required for TL DR generation , however it still stands the chance of missing out any information that is present in the full text.

We would be unable to incorporate the AIC completely to a standard pretrained model let alone the information from the full text. Hence in order to incorporate the information from the full text as well, we utilize knowledge graph that is able to capture the relations and inter relations between the key concepts that is present in the document.

We can clearly observe in fig2.2 that the source token count averages around 5,000 , which more than double the maximum token count of the AIC dataset. The full text

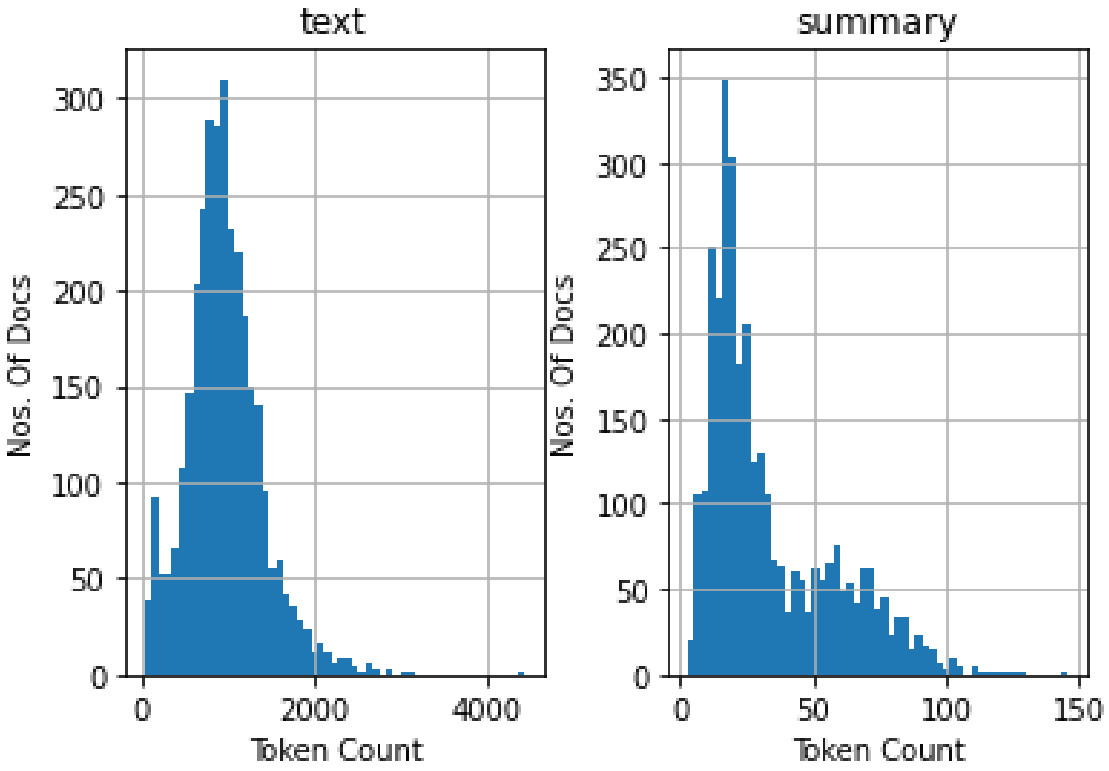

Figure 2.1: Token Length Distribution Of Source and target document-AIC

dataset has a maximum token count of 15, 000.

On stark contrast lies the token count of the target with maximum token count of 150 with the most number of documents having a token count of about 50. Hence we can clearly understand that our task requires high compression of information.

In order to judge the contextual information present in the document , we found how many of the sentences in the source document entail each other. In case a sentence entails the other, it would imply that the information present in one of the sentence is already contained in the other. The probability of entailment was found by using standard pretrained model of facebook/bart-large-mnli. Mean of the sentence wise entailment was taken and plotted for the train set of AIC dataset as seen in 2.3.We can observe that across the train set , most of the sentences entail each other with the probability of 0.3. Hence it can be said contextual information is important for scientific documents.

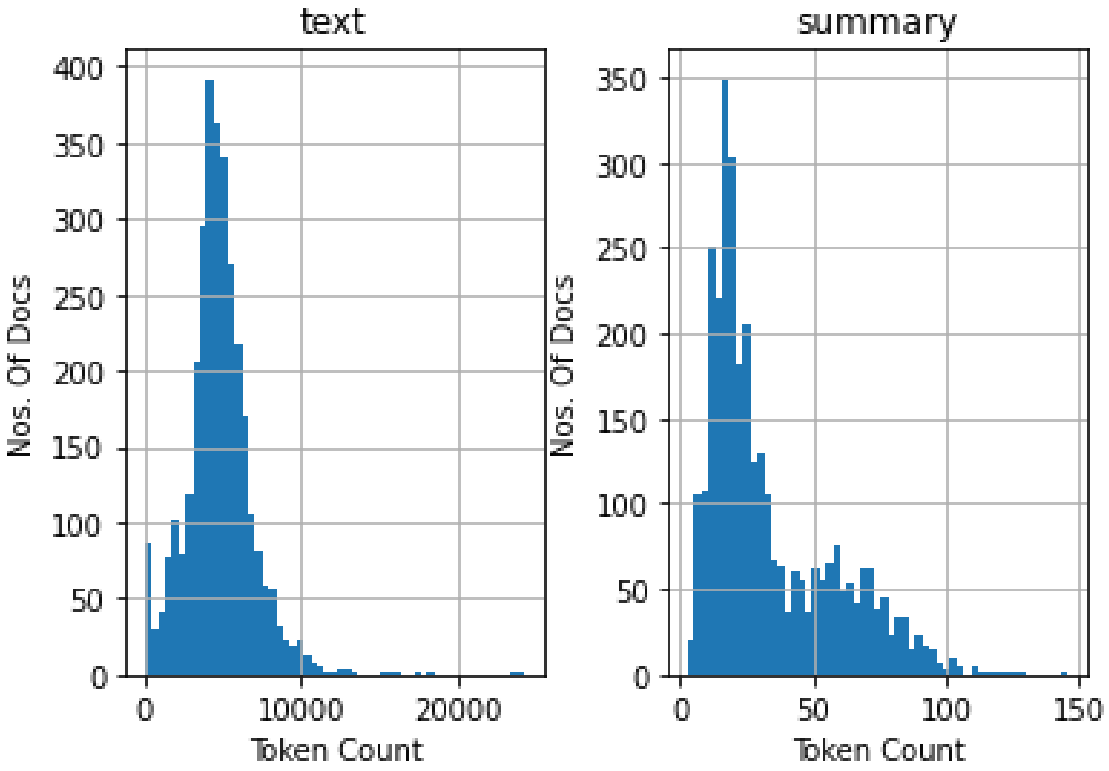

Figure 2.2: Token Length Distribution Of Source and target document-Full Text

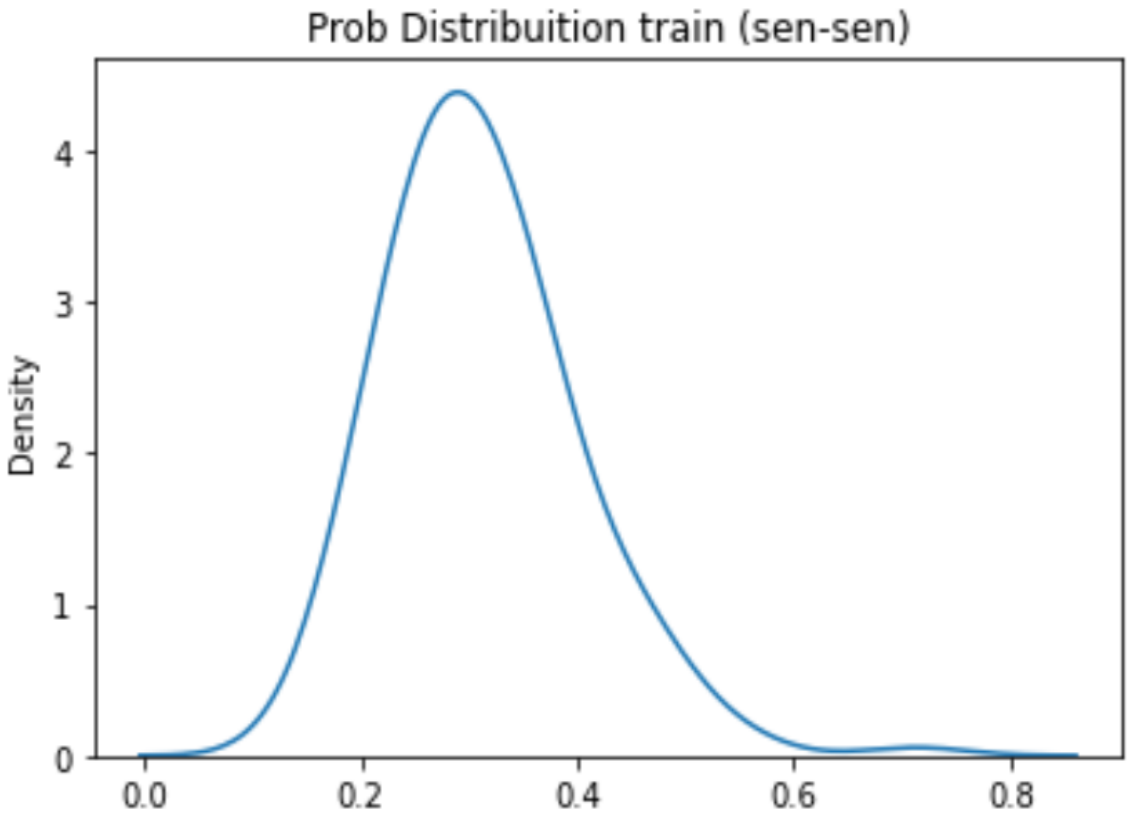

Figure 2.3: Sentence wise entailment probability distribution

Chapter 3

Proposed System

Current state of the art sequence to sequence models has token count limitation of 1024 or 512, while the token count of our full text document has a maximum of 10,000 token count. Clearly the pretrained models which have been used to produce state of the art performance fails to capture the entire context present in a scientific document.

Since context is of utmost importance in a scientific document as it is basically an interconnection of concepts, hence we believe loss of information via truncation will lead to much poor performance.We incorporate a full text knowledge graph in order to capture the inter relations between concepts presented in a scientific document and to incorporate as much information as possible for the TL DR generation.

Complex documents which spans across multiple concepts and has interrelations between them often have latent hierarchical structure present in them. Embeddings from the encoder of a standard encoder decoder architecture is said to capture the contextual information present in the input sequence. However the embeddings are in the euclidean space and distance representation in the 2-D space fails to capture the correct relationships correctly as more the hierarchy becomes more and more complex. For intuitive understanding, we can imagine a tree like structure representing hierarchies, as more and more children of the root is generated, the space to represent them properly gets limited.Due to the constraint in the space, the children of children of root nodes comes closer to the root nodes itself. Thus representation of such a structure in 2-D space would give the wrong notion of distance of the root node being closer to the children of children. Thus the projection of the 2-D distance to the hyperbolic space more specifically the Poincare ball , helps us capture the correct notion of distance. Hence each of the intermediate representation from the encoder of a standard encoder decoder model is transformed into the hyperbolic space , and incorporated along with the standard representation in the 2-D space to correctly capture the sequential contextual

information and to correctly represent the hierarchies.

3.1 Knowledge graph construction

The goal of the knowledge graph was to capture the key concepts that is presented in the research paper while incorporating the full text information. Dependency parsing was chosen as an effective method to identify the concepts present in the text.Dependency parsing is a method used in natural language processing to find meaningful connections between words in a phrase. To discover the syntactic relationships between words, dependency parsers are employed to map the words in a phrase to semantic roles. An established method for surface-level syntactic analysis of natural language documents is dependency parsing. By examining the syntactic connections between words and determining the syntactic category of each word, this method allows the syntactic structure of a sentence to be retrieved from a linear sequence of word tokens. The following steps were followed for knowledge graph generation :

1. **Co reference Resolution** : The task of locating all expressions in a text that refer to the same thing is known as co reference resolution. For many higher level NLP tasks involving natural language understanding.It will correctly help in identifying the source and target entity in each sentence. Co reference resolution was done using spacy Coreferee library which uses the current state of the art co reference resolution model Neuralcoreff for coreference resolution.

2. **Complex sentence to simple sentence** : As their name suggests, compound-complex sentences are the most difficult to understand. At least two independent clauses and one dependent clause are present in a compound-complex sentence. Simply put, a dependent clause cannot stand alone as a sentence, whereas an independent clause can. Since a complex sentence has multiple root verbs, it becomes difficult to extract the entities via dependency parsing.The following method was undertaken in order to complex sentence simplification :

   (a) Find the root of the sentence.This is the verb in the independent phrase as identified via dependency parsing.

   (b) Find other verbs in the sentence i.e those which have the root as their ancestor in the dependency parsing tree.

   (c) Find token spans for each verb by simply spanning to the nearest non verb token in the left to the nearest non verb token on the right.

(d) Return the sentence clauses as identified in the above step.

3. **Get Entities and Get Relations** :

   (a) **Finding entities** : For each of the simple sentence identified in the prev step the subject and the object clauses are identified using dependency parsing and denoted as entities.

   (b) **Finding relations** : The following as identified by dependency parsing is considered to be relations.

      i. Root verb (note we only have simple sentences with one root verb)

      ii. Adjectives

4. **Graph construction** : For construction of the graph G(V,E), where V stands for the set of vertices and E stands for the sets of edges, each of the entities and relations identified in the above step is considered to be a vertex. For each relation identified in the above step , we make an un directed edge from the associated source entity to the relation and another un directed edge from the relation to the target entity. Hence for 'n' number of relations identified we make 2n edges.Fig3.1 is a visualization of the constructed knowledge graph for one of the source document.

3.2 Hyperbolic space transformation

Complex symbolic texts often has inherent hierarchical structure present in them. Traditional methods of representations of such texts project the texts in euclidean vector spaces, which do not take into account the inherent hierarchical relationships. Projecting the representation vectors to hyperbolic dimension, most specifically the poincare ball, helps us capture the hierarchical structure and thus helps us to do correct representation of the textual information. As can be intuitively understood by the figure in 3.1 our dataset does have inherent hyperbolic structure where there is exponential volume growth of nodes with increase in the tree depth as seen by the clustering of nodes on the edges. Poincare embeddings allows us to create hierarchical embeddings in a non-euclidean space. The vectors on the outside of the Poincare ball are lower in hierarchy compared to the ones in the center.

Before jumping into hyperbolic embeddings, let us first discuss the limitations present in embeddings of euclidean metric spaces. Given a set V of m vectors (points in R^n , the Gram matrix G is the matrix of all possible inner products of V, i.e.$g_{ij} = v_i^T v_j$. A gram matrix can only be said to be euclidean if it is a positive semi definite matrix

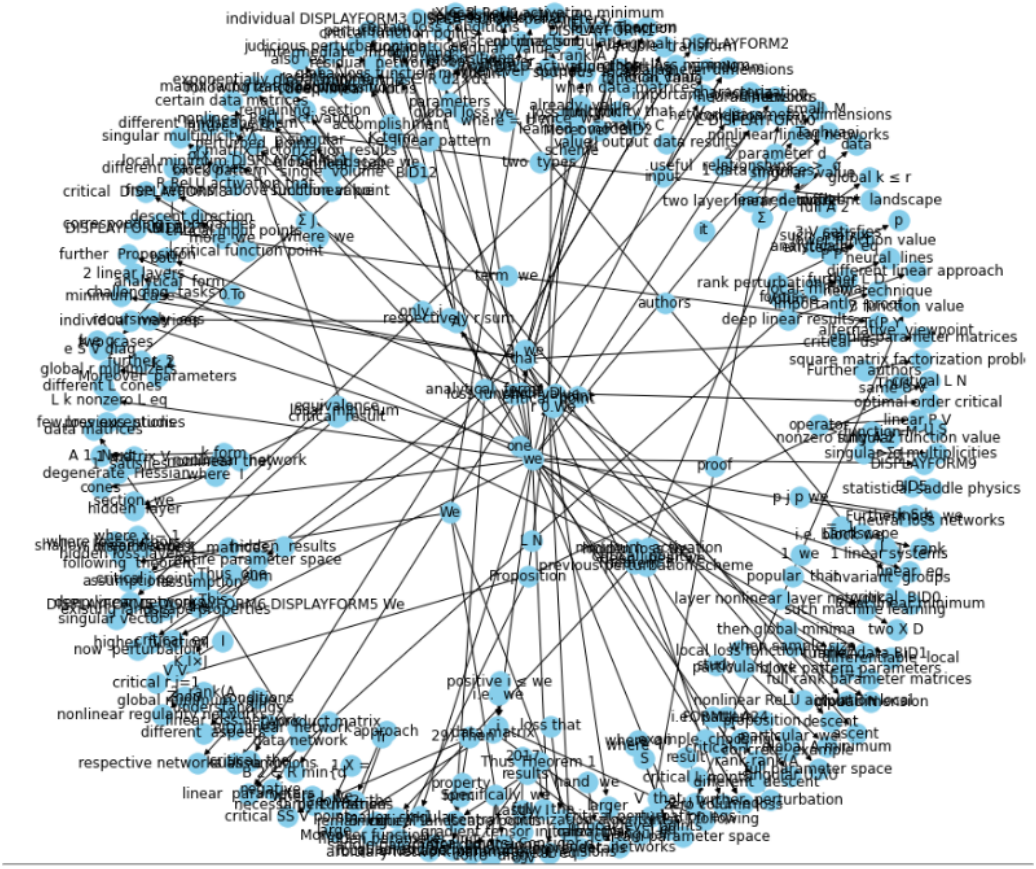

Figure 3.1: Visualization of constructed knowledge graph from a sample doc

i.e it does not have a single negative eigen value. Hence a euclidean embedding cannot perfectly reconstruct the data if the underlying gram matrix is not positive semi definite. However such data can be correctly represented in the hyperbolic space. The transformation to map a Euclidean metric tensor to a Riemannian metric tensor is an open d-dimensional unit ball is given below :

$$g_x = (\frac{2}{1 - \|x^2\|})^2 g^E$$

Our approach is to thus transform each of the intermediate representations that is obtained from the encoder to that of the hyperbolic space and in order to compensate for the loss of information that might take place in transformation, we add the transformed representation and the original representation to be fed into the sequence to sequence model

3.3 Final architecture

The final architecture thus incorporates both the components in order to do effective abstractive summarization.

The final architecture is described in the fig 3.2. The knowledge graph is constructed using the methods as described above.In order to get the graph embeddings , we use node2vec. Node2Vec is a useful embedding methodology since

1. Node2Vec clusters similar nodes to together. NodeVec embeddings for similar nodes are close together.Since our nodes are entities in the text and they are connected via dependency links, nodes belonging to the same topic group would be clustered together.

2. Random walk embeddings in our knowledge graph will give the idea of which topic is nearer or farther from each other.

We choose BART pretrained on the XSsum dataset as our base sequence to sequence model , as it has performed the best amongst our baselines.

We apply poincare normalization to each level of input representation. Both the transformed representation and the representation in the eucledian space is fed forward through the model.

The node2vec captures the graph embeddings in the euclidean space and hence poincare normalization is also applied to it.

3.4 Fusion mechanism

We treat the full text graph and the text input as two different modalities and hence take inspiration from multi modal fusion techniques.

The graph is made to query the text modality and also the text modality queries the graph for appropriate value. We hypothesize the graph query text value vector to be more important and wanted to append it across the text query graph value vector. In order to achieve the same , we flattened and did linear transformation and repeated it along the second axis for successful addition.

Further linear layers are placed between the output of the encoder and the input of the decoder , in order to eliminate any noise and to facilitate further learning.

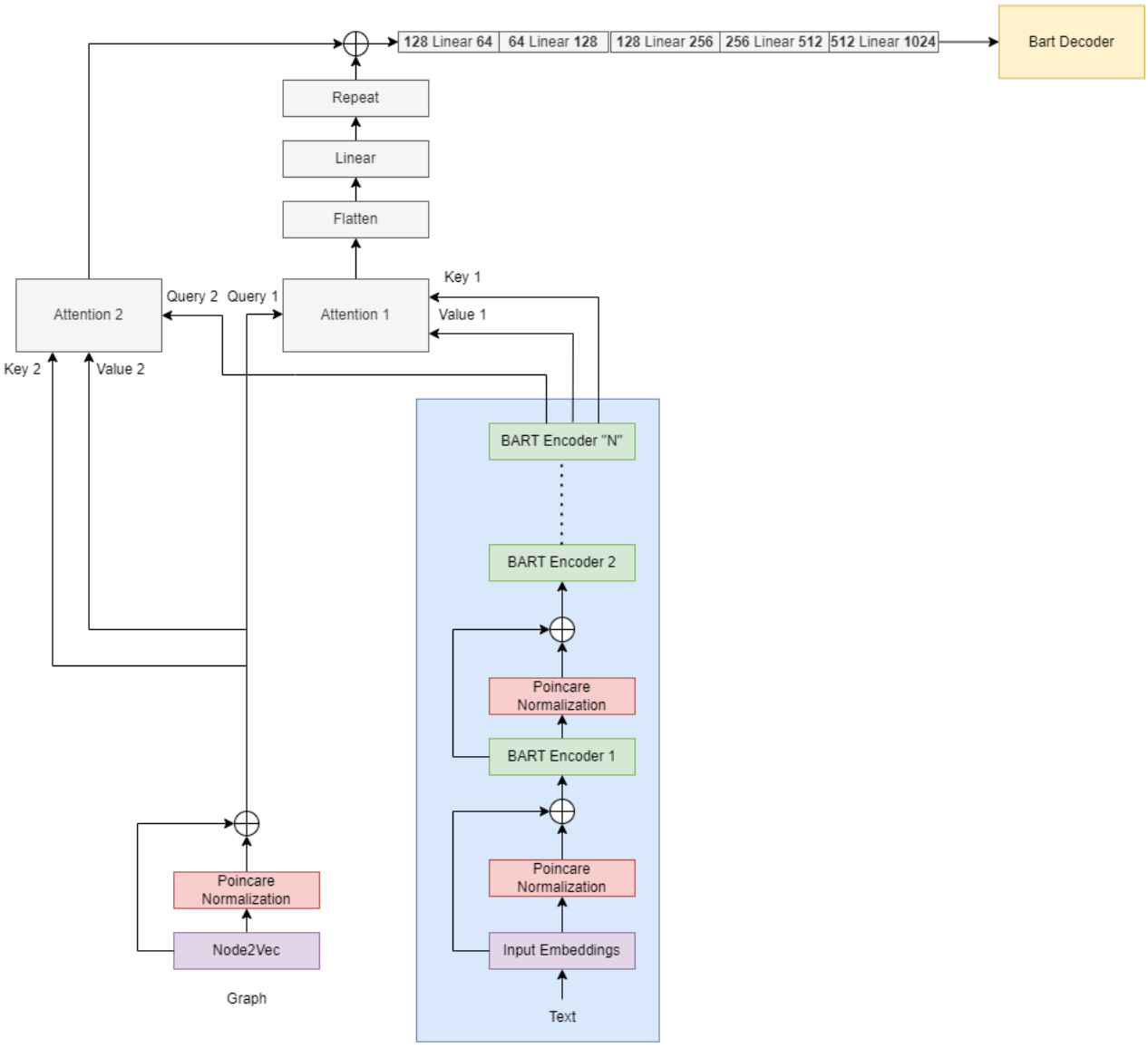

Figure 3.2: Knowledge graph assisted hyperbolic BART(**KAHB**)

Chapter 4

Experiments

4.1 Extractive baselines

First of all an extractive oracle was implemented as an upper bound, here the sentence with the highest rouge score with that of the target gold summary was returned. Since we wanted to evaluate the possibility of incorporating knowledge graphs , we evaluated with LexRank. For our extractive summarization method, we used **LexRank**, which visualizes our text as a graph and then proceeds to identify the centroid sentences in our document. It consisted of the below steps:

1. Represent the text as a graph edge weight matrix
2. Rank the sentences using the LexRank algorithm
3. Return the top five ranked sentences as the summary of the document.

We consider each sentence as node or vertex V in our graph, and our text is visualized as a completely connected graph, where there is N number of sentences and thus N^2 number of edges between them. In order to give weights to each of the edges, we calculated the cosine similarity between their embeddings of them. The sentences were embedded using the pre-trained model "MpNet". The pre-trained model has been evaluated to perform very well in sentence similarity tasks and thus was our ideal candidate. It uses the objective of masked and permutated language modeling in its pretraining.

The importance of each node i.e sentence in our case is determined by the eigenvector centrality and the via "power iteration", we calculate the eigenvector centrality score of the graph, till it converges. The power iteration increases the importance of a node,

as the importance of its neighbor increases. Hence once we have completely ranked our sentences, we return the top-most ranked sentence as our summary. On qualitative evaluation, we observed that the extractive summary is not able to capture the entire information that is present in our document since no one sentence in our entire doc can be said to capture the entire information. Hence we looked into the abstractive methodology

4.2 BART

BART(Lewis et al., 2020), has been evaluated to be able to capture long-range dependencies and is able to perform superior in the task of summarization across multiple domains. Choosing BART as our candidate, we fine-tuned the model firstly on the preprocessed source articles.We specifically choose BART pretrained on the XSum dataset as the model is already trained to do extreme summarization.

4.3 Bert2GPT and Bert2Bert

As described in the methodology by (Rothe et al. ,2020), we employed bert as our encoder while using GPT as the decoder architecture. We know that large pre-trained standalone encoder and decoder models that are freely available checkpoints, like BERT and GPT, can improve performance and lower training costs for many NLU tasks. We also know that encoder-decoder models are essentially a combination of standalone encoder and decoder models. This naturally raises the questions of how encoder-decoder models can benefit from stand-alone model checkpoints and which model combinations work best for specific sequence-to-sequence tasks. The work by (Rothe et al., 2020) provides an excellent examination of several encoder-decoder model pairings and fine-tuning methods. Following there methodology we attempt two baselines for our dataset.

However the performance of these models were not satisfactory on our dataset. This could be because of the high compression that was to be achieved in our case and loss of gradient flow in the resulting large architecture

4.4 Transformer encoder decoder

The transformer as proposed by (Vaswani et al., 2017) was implemented with four stacks of encoder decoder. The model was able to capture the frequently occurring words and phrases in the document , for ex : 'we propose', 'novel' , etc. However it fails to capture the keywords occurring at document level.

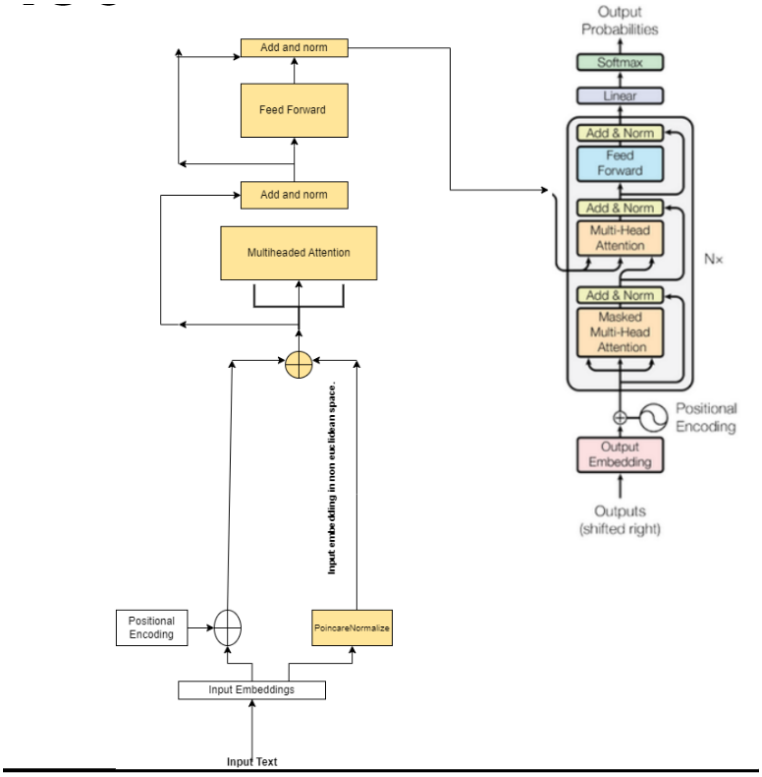

Figure 4.1: Poincare transform at input embedding level

4.5 Transformer with poincare at embedding level

We evaluated our hypothesis of transforming the embeddings to an hyperbolic space , by first applying the transform at the input embedding layer of the transformer as shown in the. This gave promising results, we were able to capture some of the keywords at the document level and was able to increase in our rouge scores. fig4.2

4.6 Transformer with poincare at intermediate representation level

Since there was promise at applying the poincare transform proving our hypothesis to be correct, we went ahead and applied the poincare transform at each of the encoder layer output of the transformer. Since each of the output of the encoder layer gives a representation of the text at different hierarchies , hence transforming the input

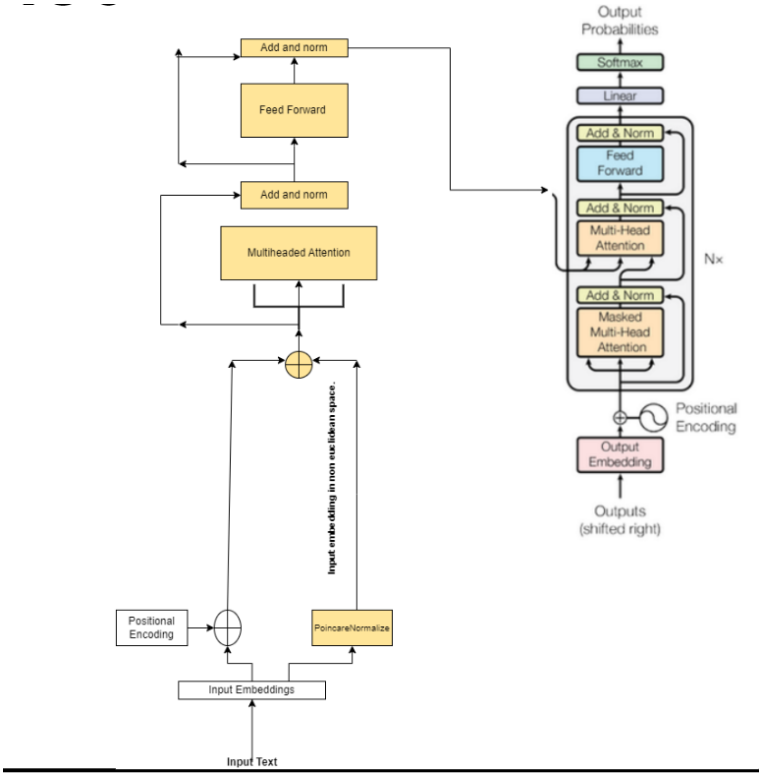

Figure 4.2: Poincare transform at input embedding level

representation at each level made intuitive sense. This hypothesis of ours were proved to be correct , as we achieved improvements in our metric of ROUGE and was also able to see improvement in the quality of the summary generated qualitatively. The architecture as described in

4.7 Transformer with graph and KG

Once we achieved the improvements with the Poincare transformation , we tried to incorporate the full text knowledge graph which was constructed with the methodology stated above. On incorporating the knowledge graph in the transformer as illustrated in the figure 4.3

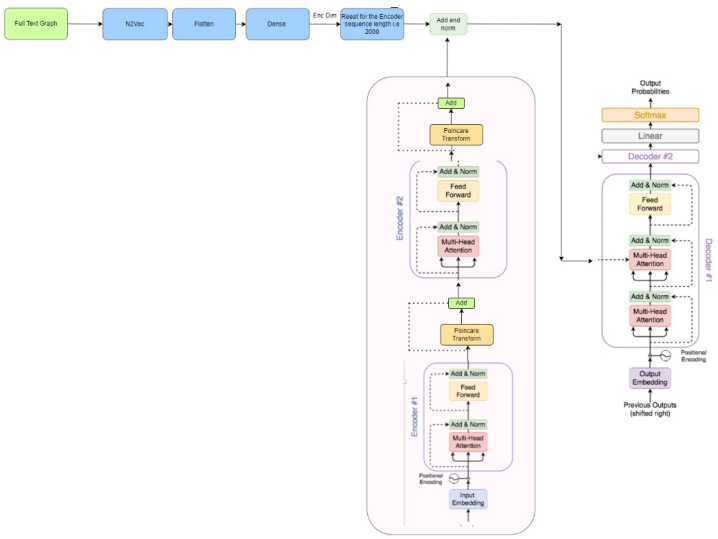

Figure 4.3: Poincare transformation of input representation and incoroporation of full text knowledge graph

Chapter 5

Results

The reported results in table 5.1 are means of the rouge scores for each instance of the test set of SCITLDR-AIC.

We can clearly see that incorporating the poincare transformation results in increase in rouge scores hence proving our hypothesis.

Incorporating the knowledge graph along with the poincare transformation yeilds comparable results.

5.1 Analysis

As seen in figure 5.1, the plain sequence to sequence BART is able to capture the essence of the document vut fails to capture the essential keywords and information. This can also be attributed to the fact that BART would truncate the token count to 1024 and would take in the entire source text as its input.

In fig5.2 , we improvement in the result , in terms of capturing the keywords and the concept that is present.

Figure 5.3 shows that it is able to capture the essential information and the concepts , however it fails to hit the gold summary exactly.

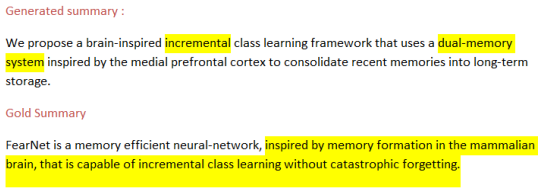

Figure 5.1: Results from BART model

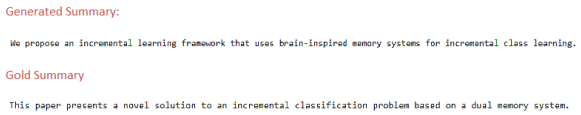

Figure 5.2: Results from Poincare BART model

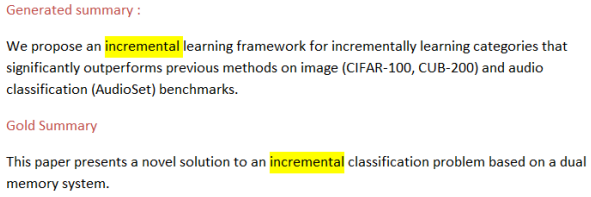

Figure 5.3: Results from KAHB

Table 5.1: Comparison of results with baselines

Model	**rouge-1**	**rouge-2**	**rouge-L**
Extractive Oracle	0.25	0.10	0.27
LexRank	0.16	0.029	0.18
Bert2GPT	0.057	0.0015	0.055
Bert2Bert	0.0007	0.000	0.0007
BART	0.345	0.154	0.306
Transformer	0.14	0.017	0.12
Transformer Poincare input embed	0.162	0.017	0.141
Transformer Poincare input representation	0.171	0.016	0.147
Transformer Poincare input representation and KG	0.177	0.018	0.152
BART Poincare input representation ↑	**0.360**	**0.161**	**0.311**
BART Poincare input representation and KG ↑	**0.363**	**0.162**	**0.314**
SCITLDR CATTS	0.317	0.111	0.250

Chapter 6

Related Works

In the extractive summarization domain, earlier works rely on rule-based systems. The rules and design features were manually designed by experts having domain knowledge. Some of the earlier abstractive summarization tasks were done based on the coherence of sentences. They mainly modified and combined the extracted phrases and sentences to generate summaries.

Neural network-based summarizers find newer representations of sentences and then find the relevance of those representations for an extractive summary. The vanilla transformer-based models have performed well in beating the earlier baselines for abstractive summaries, but they are computationally much more expensive, both in terms of time and space. The advanced architectures based on transformers have performed better and are mostly employed for this task.

PageRank is a standard algorithm for identifying essential web pages. It maps the whole web as a graph and finds the most influential nodes. LexRank, a successor to PageRank, is a graph-based method for summarizing text. It is an unsupervised paradigm that scores sentences based on the eigenvector centrality in a graph representation of sentences. It incorporates similarity metrics. The approach proposed in LexRank is insensitive to the noise present in input data arising from the imperfect clustering of sentences.

Strong results in extractive and abstractive summarization have been attained using transformer-based models.Transformer architecture[1] have been adopted for abstractive summarization tasks. The current state of the art and the most popular models used for the task of abstractive summarization are based on the transformer architecture. BART[2] trained on Xsum is the current state of the art for extreme summarization.A bidirectional and AutoRegressive Transformer(BART) is a denoising autoencoder for pretraining sequence-to-sequence models. It uses BERT encoder for

its direction agnosticism or bidirectionality and Generative PreTraining(GPT) decoder for its autoregressive feature. It is fine-tuned on summarization datasets and currently gives state-of-the-art results.

However the major disadvantage associated with the transformer based pretrained models is that it has the maximum token count limit of 512 or 1024. Hence when employed for the task of scientific documents , which well exceeds their limit of token count, there is a loss of information.

Different techniques have been previously applied for long document summarization, which includes capturing incorporating a hierarchical encoder as in the work by Cohan et al.[3]. However capturing long range information still remains a challenge. Knowledge graph based technique have also been previously explored in [4]Huang et al 's paper, which proposes to capture the semantic information present in the document by the use of knowledge graph and also comes up with a reward based system to better capture the entity interactions. The work by Nickel et al[5] brings forward the deficiency of only representing embeddings in the euclidean space. However using hyperbolic representation in the task of summarization have not yet been explored to the best of our knowledge.

Chapter 7

Conclusion

In this work we explore the use of knowledge graph and hyperbolic embeddings in order to do extreme summarization of scientific documents. With the use of knowledge graph built from the full text of the scientific document, we incorporate the full text information which is lost when using current state of the art sequence to sequence models.

We hypothesized that in order to represent the latent hierarchical information ,vector representation in euclidean space is not sufficient. Hence we transform our intermediate representations to the hyperbolic space and achieve increment in our metric. Our work is able to achieve improvement over the baseline SCITLDR-CATTS which is proposed over the dataset SCITLDR on which we evaluate our results.

We have not been able to achieve significant improvements by incorporating the knowledge graph, even though it contains the information of full text. This might be caused due to inefficient and lossy fusion methods and/or inefficient representation of graph embeddings.

Hence for future work, we would explore more fusion methodologies and different graph representation techniques.

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is All you Need."

[2] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," Oct. 2019, number: arXiv:1910.13461 arXiv:1910.13461 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1910.13461

[3] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents," May 2018, number: arXiv:1804.05685 arXiv:1804.05685 [cs]. [Online]. Available: http://arxiv.org/abs/1804.05685

[4] L. Huang, L. Wu, and L. Wang, "Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 5094–5107. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.457

[5] M. Nickel and D. Kiela, "Poincar\'e Embeddings for Learning Hierarchical Representations," May 2017, number: arXiv:1705.08039 arXiv:1705.08039 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1705.08039