

Kennesaw State University

DigitalCommons@Kennesaw State University

Master of Science in Computer Science Theses

Department of Computer Science

Spring 5-18-2020

Graphical Representation of Text Semantics

Karl Kevin Tiba Fossoh

Follow this and additional works at: https://digitalcommons.kennesaw.edu/cs_etd



Part of the [Graphics and Human Computer Interfaces Commons](#), [Numerical Analysis and Scientific Computing Commons](#), and the [Theory and Algorithms Commons](#)

Recommended Citation

Tiba Fossoh, Karl Kevin, "Graphical Representation of Text Semantics" (2020). *Master of Science in Computer Science Theses*. 43.

https://digitalcommons.kennesaw.edu/cs_etd/43

This Thesis is brought to you for free and open access by the Department of Computer Science at DigitalCommons@Kennesaw State University. It has been accepted for inclusion in Master of Science in Computer Science Theses by an authorized administrator of DigitalCommons@Kennesaw State University. For more information, please contact digitalcommons@kennesaw.edu.

GRAPHICAL REPRESENTATION OF TEXT SEMANTICS

A thesis presented to
the faculty of
the Department of Computer Science

by

Karl Kevin Tiba Fossoh

In partial fulfillment
of the requirements for the degree
Master of Science in Computer Science

Kennesaw State University

2020.05

GRAPHICAL REPRESENTATION OF TEXT SEMANTICS

Approved:

Dr. Dan Lo

Professor

Dr. Coskun Cetinkaya

Department Chair

Dr. Jon Preston

Dean

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of this thesis which involves potential financial gain will not be allowed without written permission.

Karl Kevin Tiba Fossoh

Notice to borrowers

Unpublished theses deposited in the Library of Kennesaw State University must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this thesis is:

Karl Kevin Tiba Fossoh
1106 Harbor Pointe Parkway
Sandy Springs, GA 30350

The director of this thesis is:

Prof. Dan Lo
369 J Building, 1100 South Marietta Pkwy
Marietta, GA 30060

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested Kennesaw State University

Name of user Address Date Type of use (examination only or copying)

GRAPHICAL REPRESENTATION OF TEXT SEMANTICS

An Abstract of

A thesis presented to
the faculty of
the Department of Computer Science

by

Karl Kevin Tiba Fossoh

In partial fulfillment
of the requirements for the degree
Master of Science in Computer Science

Kennesaw State University

2020.05

ABSTRACT

TIBA FOSSOH, KARL KEVIN, M.S., May 2020, Computer Science

GRAPHICAL REPRESENTATION OF TEXT SEMANTICS (49 pp.)

Director of Thesis: Dr. Dan Lo

A text is a set of words conveying a particular semantic based on their order, representation and structure. Those elements can be associated through a different set of interpretations, based on frequency and proportionality. The problem with context is that numbers do not help understand the semantics and fall short to convey the message of the text. The graphical representation of text semantics focuses on the conversion of text to images. Contrarily to word clouds that simply produce frequency mapping of words within the text and topic models that essentially give context to word frequencies and proportionalities, images keep intact the semantic and the context of the words in the text. They provide a deeper understanding and can be better interpreted. Models such as AttnGAN already exist to convert text into images with a certain level of success, but there has not been work done concerning the conversion of long and complex texts in an image or a set of images. The goal of this analysis is to first, provide an understanding of how we divide the text in bits that improve the resulting image and how does the summarization methodology affect the image result.

GRAPHICAL REPRESENTATION OF TEXT SEMANTICS

A thesis presented to
the faculty of
the Department of Computer Science

by

Karl Kevin Tiba Fossoh

In partial fulfillment
of the requirements for the degree
Master of Science in Computer Science

Kennesaw State University

2020.05

DEDICATION

*To my admirable parents that always supported me in all of my choices
and made me the man I am*

ACKNOWLEDGMENTS

The definition of a valuable research most of the time resides on the capacity to decipher the said and done and look where someone has not looked yet. This research emanated from a deep passion with new technologies and the recent improvements in term of Generative Adversarial Networks. I would not have had such a fulfilling experience without the guidance of Dr. Dan Lo and the expertise of Pr. Jose Garrido. They both provided me great insight on where to invest my energy and get great results and I am grateful for that. Thanks to my committee members for the Support, Dr Selena He and Dr Jungabb Son. Thanks to Dr Dan Lo, for whom I was an Graduate Research assistant and who brought me into the research field. Thanks to the graduate college for helping me support my studies through the Graduate Research Assistant Program. Special Thanks to Kennesaw State University for providing an adequate environment for this research and all the necessary resources to make a significant progress.

TABLE OF CONTENTS

| | Page |
|--|------|
| Abstract | 6 |
| Dedication | 2 |
| Acknowledgments | 3 |
| List of Tables | 6 |
| List of Figures | 7 |
| List of Acronyms | 8 |
| 1 Chapter I: Introduction | 9 |
| 1.1 Problem Statement | 9 |
| 1.2 Purpose of the research | 10 |
| 2 Chapter II: Literature Review | 11 |
| 2.1 Semantics | 11 |
| 2.2 Text Mining | 13 |
| 2.2.1 Text Preprocessing | 13 |
| 2.2.2 Text Cleanup | 16 |
| 2.2.3 Tokenization | 16 |
| 2.2.4 Part of speech tagging | 16 |
| 2.3 Text Summarization | 17 |
| 2.3.1 Extractive Text Summarization | 18 |
| 2.3.2 Abstractive Text Summarization | 19 |
| 2.3.3 Text Coherence | 20 |
| 2.4 Natural Language Processing | 21 |
| 2.4.1 TF-IDF | 22 |
| 2.4.2 LDA | 22 |
| 2.5 Machine Learning | 24 |
| 2.5.1 Neural Networks | 24 |
| 2.5.2 Convolutional Neural Network | 26 |
| 2.5.3 LSTM | 28 |
| 2.5.4 Generative Adversarial Network (GAN) | 30 |

| | | |
|-------|---|----|
| 3 | Chapter III: Methodology | 31 |
| 3.1 | Dataset | 31 |
| 3.2 | Models | 32 |
| 3.2.1 | Attention-Based LSTM | 32 |
| 3.2.2 | Seq2Seq | 33 |
| 3.2.3 | Bidirectional LSTM | 34 |
| 3.2.4 | Attention-based Bidirectional LSTM | 34 |
| 3.2.5 | TextRank | 34 |
| 3.2.6 | Attentional Generative Adversarial Network (Attn-GAN) | 35 |
| 3.3 | Preprocessing | 35 |
| 3.4 | Environment | 36 |
| 4 | Chapter IV: Result | 39 |
| 4.1 | Data Understanding | 39 |
| 4.2 | Models Setup | 39 |
| 4.3 | Models Results | 40 |
| 4.3.1 | Models Validation and Training Loss curves | 40 |
| 4.3.2 | Coherence results and LDA results | 41 |
| 4.3.3 | GAN Images | 41 |
| 5 | Chapter V: Conclusion | 45 |
| | References | 46 |

LIST OF TABLES

| Table | Page |
|---|------|
| 2.1 Semantic association | 12 |
| 2.2 Lowercase transformation | 14 |
| 2.3 TF-IDF Table | 22 |
| 4.1 Hyper-Parameters | 40 |
| 4.2 Rouge scores for our different summarization models | 40 |

LIST OF FIGURES

| Figure | Page |
|--|------|
| 2.1 Tokenization models | 17 |
| 2.2 Summarization Overview | 18 |
| 2.3 Abstractive Text Summarization | 19 |
| 2.4 Plate Representation of LDA, From [SB19] | 25 |
| 2.5 Node with inputs, From [Pre16] | 26 |
| 2.6 Convolutional Neural Network, From [Sum18] | 27 |
| 2.7 Sample LSTM, From [dpr19] | 29 |
| 2.8 Generative Adversarial Network, From [XZH ⁺ 17] | 30 |
| | |
| 3.1 Overall System | 32 |
| 3.2 Attention Based LSTM, From [Nir19] | 33 |
| 3.3 Seq2Seq, From [Cha19] | 34 |
| 3.4 Bidirectional LSTM, From [Lil19] | 35 |
| 3.5 Attention based Bidirectional LSTM, From [ZST ⁺ 16] | 36 |
| 3.6 Attention Based GAN, From [XZH ⁺ 17] | 37 |
| 3.7 Example of generated image, From [XZH ⁺ 17] | 38 |
| | |
| 4.1 Summaries and articles lengths histogram | 39 |
| 4.2 Training and validation loss for Attn-LSTM | 41 |
| 4.3 Training and validation loss for Seq2Seq | 42 |
| 4.4 Training and validation loss for Bidirectional LSTM | 42 |
| 4.5 Training and validation loss for Attn-Bidirectional LSTM | 42 |
| 4.6 Summaries Coherence histogram | 43 |
| 4.7 Articles Coherence histogram | 43 |
| 4.8 Generated images | 44 |

LIST OF ACRONYMS

GAN - Generative Adversarial Network
LDA - Latent Dirichlet Allocation
RNN - Recurrent Neural Network
CNN - Convolutional Neural Network
LSTM - Long Short Term Memory
Attn - Attention
IR - Information Retrieval
S2S - Sequence-2-Sequence
TF-IDF - Text Frequency - Inverse Document Frequency
POS - Part Of Speech
NLP - Natural Language Processing

LDA - Latent Dirichlet Allocation

1 CHAPTER I: INTRODUCTION

1.1 Problem Statement

Complex sentences surround us every day with various meanings and content. They are one important part of our language and represent the vast majority of our readings. Complex sentences are composed of an independent clause and one or more dependent clauses. Along with simple and compound sentences, they form the main sentences structures we can find in books, speeches and other elements using text. In the books we read sometimes are illustrations. These illustrations generally take 5 weeks to complete for a 24 to 32 pages books and can go up to 5 months for the most complex illustrations. With today's technology, generating a specific image of a bird with specific descriptive elements is done in a matter of seconds using a Generative Adversary Networks. Uplifting such a task from months to minutes would be a dream but this ideal solution is still far from our reality. Nonetheless, the capacity to build a system that could provide visual reference from sentences found in a book could drastically improve readability for young kids, hence the scope of this research. One assumption that would come is why not simply feed a GAN with the book and wait for a result. The problem with GANs, and specifically those related to text-to-image conversion, is the problem of semantic separation, sentence logic, and at the ground base the fact that GANs cannot properly digest thousands of lines of text at once. Also, abstract elements cannot be represented as images as they simply do not have any physical representation. These different variables coming from the text make its conversion into image a real challenge. Throughout our research, we establish as a first step understanding complex sentences and providing a way to reduce them to summaries, then convert them into logic groups that can then be ingested by GANs.

To assist ourselves in our task, we decided to compile different novel summarization algorithms along with logic-based association algorithms and GANs. Our goal is not to

reinvent the wheel but rather use and tweak the tools already present to lift a new solution. This is why in our first iteration of the process, we assessed which summarization algorithm worked best on an average of different summarization datasets, before taking a deeper dive with the best two summarization algorithm and apply the logical separation algorithm based on topic relatedness. The novel structure implemented in this work is the final result of the summarization and logical association that we decided to call “seed” for the purpose of this research. Each seed can become an image, or a set of images based on the length and relatedness of the sentences. We also raise the hypothesis that the coherence of a seed impacts the quality of the image. This is verified through testing the different seeds from the two provided processes along with the two used GANs. Our end goal is the generation of an illustration set that follows the logic and continuity of the text provided.

Based on the knowledge acquired through readings and preceding experiments, we decided to apply the techniques provided and establish a model for graphical representation of text semantics. Rather than building our own structure, we combine different novel models that proved to be more efficient than commonly used models. The hypothesis is that the average coherence of a text can be increased by associating its logical groups. We also set as hypothesis that the image provided by the most coherent seeds will be better than the image provided by the less coherent ones.

1.2 Purpose of the research

The purpose of the research is to assess the capacity to define a usable pipeline to the transformation of long complex text into set of images and establish which methodology provides the best results in term of image realism. We have to consider that the image realism is a pretty subjective topic and does not correlate to a mathematical operation. The nature of this work is qualitative but is supported by clear metrics surrounding its foundations.

2 CHAPTER II: LITERATURE REVIEW

2.1 Semantics

Semantics is the study of the meaning of words, phrases and sentences [Bar73]. The study of semantics is centered around the meaning of words and sentences. Two types of meaning are to be observed when discussing the meaning of words and sentences. The conceptual meaning refers to the basic meaning of the word conveyed by the language.

e.g. : Foot: the lower extremity of the leg below the ankle, on which a person stands or walks

In this case the word finds a singular meaning that is understood as base. The associative meaning deals with the connotation of the word. The meaning the word varies based on the other elements of the sentence surrounding it.

e.g. : This stool is one foot tall and can be used as support

In this case, the word foot is referred as a unit of length/height because of the surrounding elements of the sentence. The English language reveals oddity that can help understand what is the meaning of a word based on the words it is and what it is not. Relatively to the combinations chosen, we can notice some sentence structures do not work:

e.g. : The throne sat on the king

Even though the sentence can be pictured, it appears odd as it is not common. The sentence is syntactically good but semantically odd. The components of conceptual meaning are general notions that appear in speech or thought. Elements can be associated with specific capacities. In the example presented above, the throne cannot sit on the king, generating an oddity within the sentence. Sitting is not an active property of a chair, but a human has an active property that is sitting. We can therefore construct the conceptual meaning of a word based on its properties and counter properties.

| | Throne | King | Man | Sun | Ground |
|-------|--------|------|-----|-----|--------|
| Walk | - | + | + | - | - |
| Sit | - | + | + | - | - |
| Male | - | + | + | - | - |
| Shine | + | - | - | + | - |
| Soil | - | - | - | - | + |

Table 2.1: Semantic association

Based on the table above, we know that a king can be a male but cannot be a soil, and a man can sit, whereas the sun cannot sit but can shine. Such a table helps understanding the conceptual meaning of an entity, and can be used as a reference to grasp the coherence of a statement. From the analysis gathered in [Kuz15], the composition of the components of the language is a system of basic concepts, states, place and properties, a system of relationships, a system of spatial concepts, a time line, a set of causal relationships and a goal. We have to remember that these elements are more qualitative than quantitative, making it difficult to design a formal process to capture them and define a language processor. Multiple factors impact the semantic of a text as stated in S. Pinker's book Introduction to special issue of Cognition on lexical and conceptual semantics [LP91]. The creation of methodologies surrounding language cannot grasp those qualitative elements properly, hence why the quantitative analysis of language has been the main train of research in the recent years. Semantics reveals to be a qualitative attribute [FvKS04] that is deeply rooted within the context, state of mind and power dynamic of the discussion. Nonetheless, tools such as LSA [Lan06] claim to create vector-based representations of text which state to be able to capture semantic content.

2.2 Text Mining

From the definition of Kumar et al [QSF⁺18], text mining denotes the analysis of large sized natural language datasets and detection of linguistic usage patterns to extract useful information. Text mining approach relies on quantitative analysis over subjective qualitative analysis. Text mining is divided into four areas which are:

Data Mining: Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis.

Information retrieval: IR refers to the different methodologies and guides used to retrieve relevant documents from large datasets with maximal efficiency.

Natural Language Processing: It is the study of human language whose goal is to have computer understand language the way human do.

Information Extraction: It refers to the extraction of structured data from unstructured or semi unstructured machine-readable documents [6]. Text mining involves fundamental steps that ensure the efficiency of the mining. These steps are the text preprocessing, the text clean up, the tokenization and the part of speech tagging.

2.2.1 Text Preprocessing

Text preprocessing involves all the modifications that convert the text in a form that is predictable and analyzable. The specificity of this task are affected by the approach and the domain. The approach refers to the methods that will be applied to the text as data, while the domain refers to the area of influence of the text, or the nature of its content.

e.g. Task = Sentiment Analysis [approach] + Amazon Reviews [domain]

In this example, the task we have at hand is to define a sentiment analysis of amazon reviews for specific products. Amazon reviews are compiled in a specific format and may contain elements that are or are not relevant to the domain, or text elements that are specific to the domain and should be given a higher importance. Sentiment analysis requires specific

elements such as adjectives and adverbs to be kept intact. By associating the approach and the domain, we are able to pre-process text in a way that improves the overall result of the next steps. This preprocessing can impact the datatype of the text data. Different preprocessing operations exist and each have a specific impact to the text and resulting output. These operations are:

Lowercasing: This operation converts all elements of the text into lower case. This avoids differentiation between elements at the beginning of sentences and enable to capture all the variations of a word under the same word. Situations may arise where conserving the case is important as some words may have different meaning depending on the case [Man09].

| Raw | Lowercased |
|--------|------------|
| Name | name |
| NAME | name |
| naMe | name |
| TOMCAT | tomcat |
| toMcat | tomcat |

Table 2.2: Lowercase transformation

Stemming: It is the process of reducing a word to a stem or root dictionary for that is used by prefixes and suffixes. Stemming is mostly used in information retrieval as it enable to find more results related to a word by using its dictionary root form, increasing the recall of the research.

e.g. :Flexible, flexing, flexibility, flexes → flex

Lemmatization: it usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma [CCP19].

Lemmatization usually provides better results in the course of information retrieval as it uses more logic rules than stemming. Lemmatization transforms the word to its actual root rather than just removing part of the word as in stemming. Most of the time, lemmatization cannot be used on top of stemming as they could infer with each other.

e.g. : Feet, Footing, footage =_i foot

Stopwords removal: Stopwords are the different words most frequently appear in the language and sometime refer to conjunctions and pronouns that do not contribute a lot to the meaning of the sentence [CCP19]. By removing low information words from tex, we can focus on the more important words. Stopwords removal is always associated with a stop word list that can more or less impact the words to be removed. Stopwords do not impact much classification systems but help reduce the sample size [QSF⁺18, WK92].

e.g. : What is the name of the president of the US? → name president US

Noise removal: It concerns removing digital pieces or numbers that can infer with the text analysis. It is highly important as some datasets have specific formats that can make any other processing obsolete [6]. It is one of the most important steps of text preprocessing and is highly domain dependent.

e.g. : [sentence] The name of tag is sentence [sentence] → The name of the tag is sentence

Text enrichment: It simply revolves around adding relevant information to the text data that was not previously present [CCP19]. Text enrichment adds more semantics to the original text, hence improving its predictive capacity. Word embedding layers are really popular as a text enrichment technique, especially for deep learning models for classification, search, summarization and text generation [HC16]. The idea behind all of the word embeddings is to capture with them as much of the semantical, morphological, context and hierarchical information as possible. Most of them are based of a vector

mapping within the text and documents. One-hot encoding, TF-IDF, Skip-gram, Word2vec, GloVe and FastText are example of embeddings commonly used for text enrichment.

2.2.2 Text Cleanup

Text cleanup generally refers to the correction of elements within the text. Operations in the text cleanup are generally encoding the data to the right format (UTF-8, ASCII), apostrophe lookup, removal of expressions (specific to theater pieces), standardizing words, grammar check and spelling corrections.

Example: I was sooooo despaired to finish this papers I “blasted” through it tho. lol
→ I was so despaired to finish this paper that I blasted through it though.

This part is generally helpful for domain specific information such as Tweets, theater pieces, narrative books and encoded text [Man09]. Text cleanup can intervene earlier in the process if required to remove word ambiguity and ease the process of lemmatization.

2.2.3 Tokenization

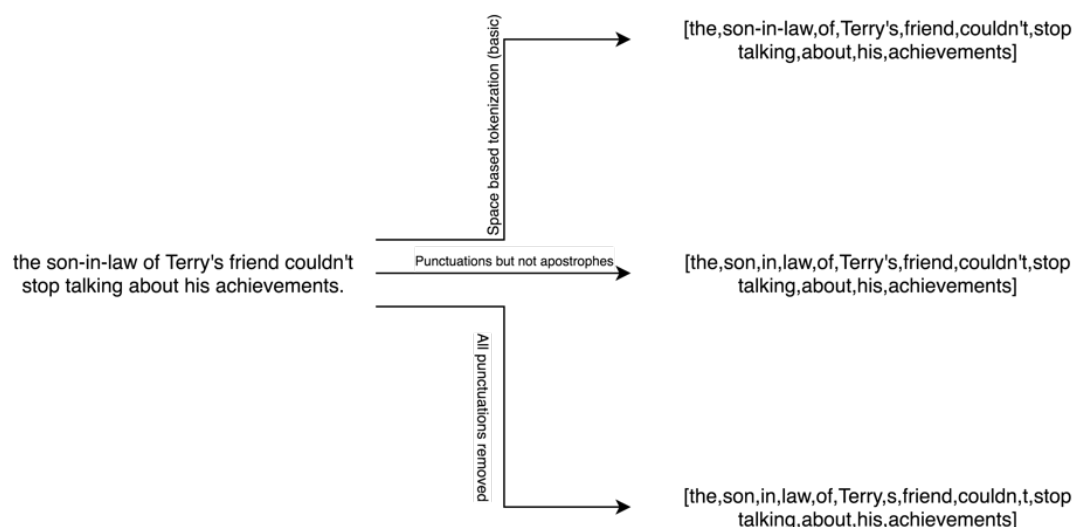
Tokenization is breaking up a text in a set of tokens, that are individual words [MS12]. The tokenization can varies based on the punctuation rules of the text. Noise removal and text cleanup make it easy to tokenize text, but compose structures requiring apostrophes or dashes can make the tokenization more complex.

Information extraction is the branch of text mining associated with the creation of summaries, topic modeling and other extractive techniques that provide significant and useful information to the user [CCP19].

2.2.4 Part of speech tagging

It is the process of word-category disambiguation during which words in the text are marked up as part of speech, based on their definition and context. In simple terms, part of speech tagging assigns a distinctive property to the word (adverb, pronoun, parent,

Figure 2.1: Tokenization models



power) that influences the understanding of the sentence for different summarization and information retrieval models.

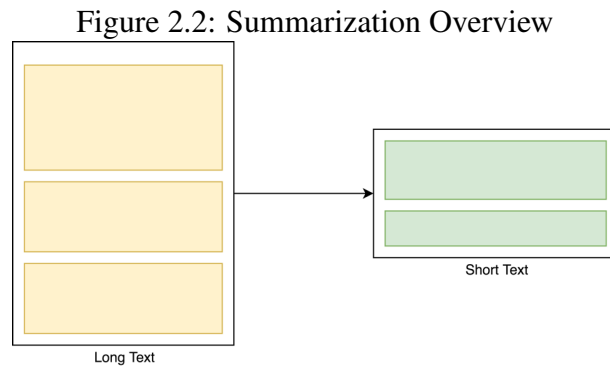
e.g. : ['Can', 'you', 'please', 'buy', 'me', 'an', 'Arizona', 'Ice', 'Tea', '?', 'It', "'s", '\$', '0.99', '.'] → [(‘Can’, ‘MD’), (‘you’, ‘PRP’), (‘please’, ‘VB’), (‘buy’, ‘VB’), (‘me’, ‘PRP’), (‘an’, ‘DT’), (‘Arizona’, ‘NNP’), (‘Ice’, ‘NNP’), (‘Tea’, ‘NNP’), (‘?’ , ‘.’), (‘It’, ‘PRP’), (‘’s’, ‘VBZ’), (‘\$’, ‘\$’), (‘0.99’, ‘CD’), (‘.’, ‘.’)]

Successfully operating those initial steps of text mining widely ensures the success of the training of the model to be defined later on. In the case of text summarization, it is shown that part of speech tagging helps in assessing what are the important parts of the text and which are not relevant, especially in extractive text summarization with sentence scoring [Sar07].

2.3 Text Summarization

Text summarization is the process of converting a text into a smaller text that contains its most relevant information. Text summarization belongs to the sub-category of text

mining called information extraction. Information extraction is the automatic extraction of structured information such as entities, relationships with entities and attributes from unstructured sources [Moh13][14]. Text summarization aims to transform long texts into short version, which can be difficult to achieve manually. Automatic text summarization refers to the operation of text summarization using a computer or machine.



The summarization of text is divided into two categories which are the extractive text summarization and the abstractive text summarization. Each uses specific factors and techniques with different level of success.

2.3.1 Extractive Text Summarization

Extractive text summary is one methodology of text summarization that consists in picking parts of a sentence and compile them as a summary based on specific metrics. Extractive text summary has been the summarization technique of choice for years before the recent improvements of machine learning and abstractive text summarization. Extractive summarization main issue resides in the coherence of the result [AA09]. In effect, sentences are sometimes longer than needed and contain information that lack the required context to efficiently represent the information at hand. Contradictory points also tend to not be caught properly by extractive text summarization [QSF⁺18, Moh13,

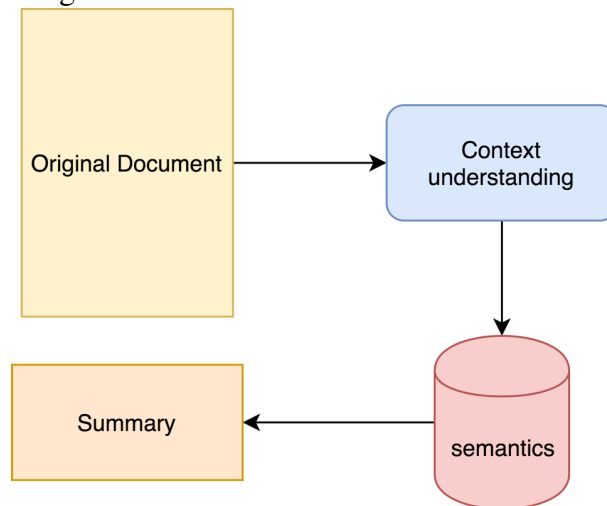
FDL+13]. Example: Joseph and Mary rode on a donkey to attend the annual event in Jerusalem. In the city, Mary gave birth to a child named Jesus. =; Joseph and Mary attend event Jerusalem. Mary birth

Using POS tagger, along with other textual metrics such as the length of the key phrase, the frequency of the key phrase, the most recurring word in the key phrase and the number of characters in the key phrase, it is possible to define an algorithm performant enough to extract accurate summaries from a text [13].

2.3.2 Abstractive Text Summarization

Contrarily to extractive summarization that crops and stitches pieces of the original document to create a comprehensive summary, abstractive text summary generates natural language summaries from a document input while retaining the important points. Abstractive text summarization most of the time uses Natural Language Processing and Deep Learning to create coherent and comprehensible summaries. Abstractive summarization method compiles words based on their semantics. The goal of abstractive summarization is to convey a text with more critical from the original text.

Figure 2.3: Abstractive Text Summarization



2.3.3 Text Coherence

Coherence is the property associated with text that are easy to read and understand [MH91]. A text is said to be coherent if there is a high degree of meaning overlapping between consecutive sentences [Lan06]]. This means that based on a certain measure of similarity of the sentences, we can observe coherence between two sentences. The closer they are in term of similarity, the more coherence there is in the textual content. Our paper focuses on semantic relatedness to provide summaries clusters before evaluation. Lexical cohesion arises from the semantic relationship between words [LH13]. Therefore, our main requirement is to have a lexical relation between the words and sentences in the text. Coherent units will have a high concentration of lexical chains [MH91]. The similarity of sentences can be based on their TF-IDF scores, making it a vector based approach. The coherence is measured through averaging the topic relatedness of consecutive sentences. The coherence of a sentence T can be expressed as:

$$coherence(T) = (\sum_1^n sim(S_i, S_{i+1})) / (n - 1)$$

Where $sim(S_i, S_{i+1})$ is the similarity between sentence i and sentence $i + 1$. For text analysis, It has been proven that cosine similarity provides the best result in term of topic grouping and clustering. A more in depth calculation system, Distance Weighted Cosine similarity, provides even better results [KB18]. Our similarity between sentences can be expressed as:

$$sim(S_i, S_{i+1}) = cos(\mu(\vec{S}_i), \mu(\vec{S}_{i+1}))$$

Where $\mu(\vec{S}_i) = \frac{1}{|S_i|} \sum_{w \in S_i} \vec{w}$ and \vec{w} is the vector for word w . Maximizing the text coherence of the summaries is the intent of our summarization. The problem is that abstractive summarization techniques do not take this in account, making summaries

sometimes incongruent [Sar07, FDL⁺13]. Extractive summarization techniques based on sentence scoring tend to have an advantage in that domain by compiling different possible scores (similarity to header, similarity to title, similarity to non consecutive sentences of the text, sentence length) and extracting the sentences with the highest scores [CY04, CAR16], and providing the high level of text coherence. Therefore, elements such as the position of the sentence, the title similarity and the sentence-to-sentence cohesion should be taken in account when defining our summaries. The ratio of sum of similarities is one of the best elements for sentence scoring as it gives a view of the coherence of one statement with all the other sentences in the document. It can be expressed as:

$$f(S_i) = \frac{SSS_i}{\max(SSS)}$$

Where SSS_i is the sum of similarities shared by the sentence i and $\max(SSS)$ is the maximal value of the different similarities sum for all the sentences in the document. Adding such features as input to our model could improve the overall coherence of the final summarized document [CAR16].

2.4 Natural Language Processing

Natural Language processing (NLP) refers to the branch of artificial intelligence related to the use of natural language between the computer and the human. In our specific case, summarization is the activity of Natural Language Processing we are trying to achieve. Beyond syntax analysis and semantic parsing, mathematics-based techniques enable us to gather more information from the documents at hand. Among those techniques we have TF-IDF and LDA.

2.4.1 TF-IDF

Text Frequency Inverse Document Frequency is a statistics-based techniques that provides point values to words within a corpus. In effect, be x the number of documents the word m appears in and w_i be the frequency of the word in a specific document.

$tf(m)$ = number of times the word m appears in the document / total number of terms in the document
 $IDF(m) = \log_e(\text{total number of documents} / \text{number of documents with the term } m \text{ in it})$

| | | | TF | TF | TF | TF-IDF | TF-IDF | TF-IDF |
|-----------|-------|-----|----|----|----|--------|--------|--------|
| Term | DF | IDF | D1 | D2 | D3 | D1 | D2 | D3 |
| Car | 18.16 | 1.6 | 27 | 4 | 24 | 44.5 | 6.6 | 39.6 |
| Insurance | 6.72 | 2.0 | 3 | 33 | 0 | 6.2 | 68.6 | 0 |
| Best | 25.23 | 1.5 | 14 | 0 | 17 | 21 | 0 | 25.5 |

Table 2.3: TF-IDF Table

TF-IDF score is essential in information retrieval and information extraction as it provides a scoring to documents based on their relatedness to a specific word. In a sense, sentence scoring [CY04] based on TF-IDF assumes that if a sentences has more specific words, then it must be more important. Also, based on ROUGE evaluation [WB11], we observe that the summarization based on text scoring using TF-IDF provides the best result in term of recall and is one of the best solution for extractive summarization.

2.4.2 LDA

Latent Dirichlet Allocation is a probabilistic topic modeling technique that enables to evaluate the topics proportions and establish relevant topics out of our document content. LDA is sometimes uses in text unsupervised classification due to the absence of precise

classification parameter [Cla13]. It appears to be a great tool when there is a need for topics and classification of elements. Each document can be described by a distribution of topics and each topic can be described by a distribution of words. LDA is a probabilistic model with a corresponding generative process from which documents are supposed to be generated [Gey07]. The generative process can be made of 3 steps which are: Randomly selecting a distribution of topics, for each word, randomly choose a topic among the distribution of topics and randomly choose a word from the selected topic. The generative distribution can be expressed using the following terms:

- $B_{1:k}$ are the topics where each B_k is a distribution over the vocabulary
- O_d are the topic proportions for document d
- $O_{d,k}$ is the topic proportion of topic k in document d
- Z_d are the topic assignments for document d
- $Z_{d,n}$ is the topic assignment for word n in document d
- W_d are the observed words for document d

The joint distribution is expressed as:

$$p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^K p(\beta_i) \prod_{d=1}^N p(\theta_d) \prod_{i=1}^K p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n})$$

LDA is based on two main probabilistic distribution that are the multinomial distribution [Lin16] and the Dirichlet distribution [Yil12]. Observing the generative process of the LDA, we can isolate two main elements:

- B_k which is the distribution over vocabulary for topic k
- $O_{d:k}$ which is the topic proportion for topic k in document d .

The second parameter is the one we are mainly concerned about as grouping the summary statements logically can be reduced to compiling similar and consecutive statements of the same topic in one logic block that we decided to name a seed. Using the posterior estimate from the Gibbs sampling [HF95], we have the following equation:

- Z_i is the topic assigned to the i th token in the whole collection
- D_i is the document containing the i th token
- W_i is the word type of the i th token
- Z_{-i} is the set of topic assignments for all the remaining tokens
- \cdot Is the remaining information such as the hyperparameters.

$$p(z_i = j | z_{-i}, w_i, d_i, \cdot) \propto \frac{C_{w_i j}^{WT} + \eta}{\sum_{w=1}^W C_{w j}^{WT} + W\eta} \frac{C_{d_i j}^{DT} + T\alpha}{\sum_{t=1}^T C_{d_i t}^{DT} + T\alpha}$$

In which C^{WT} and C^{DT} are the matrices count for the word topic and the document topic. For this equation, we can find our topic proportion per document to be given by the equation:

$$\theta_{dj} = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^T C_{dk}^{DT} + T\alpha}$$

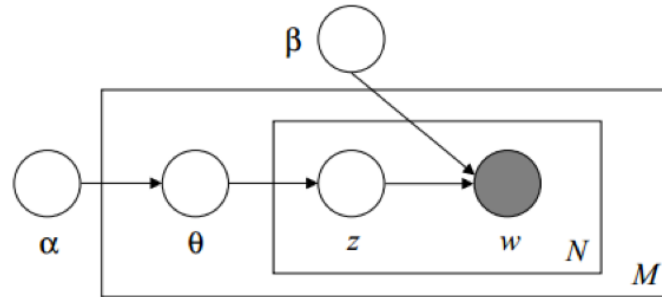
Therefore, for high values of alpha, we observe that our topic proportion tends to 0. Finding the right value for alpha can increase the chance to efficient topic allocation of our documents.

2.5 Machine Learning

2.5.1 Neural Networks

Neural networks are algorithms modeled after the logic of a human brain and set to recognize patterns that may not be represented linearly without complex transformation.

Figure 2.4: Plate Representation of LDA, From [SB19]



The patterns are numerical and contained in vectors into which any sensorial data must be translated. Neural networks are especially useful in solving problems that cannot be solved linearly or using clustering. Most of the time, the problem is too complex to be solved using the simple regression techniques [FDL⁺13, ON15]. The simplest form a neural network is a perceptron or artificial neuron. The main characteristics of an artificial neuron are the inputs and the activation function. Generally, the inputs are a set of numerical or binary data fed to the neuron in a specific order. The activation function is a switch that turns on whenever a specific threshold is reached and provide the output value, or change the state. That change of state can be from 0 to 1, from -1 to 1 or from 0 to values greater than 0. The most common function is the sigmoid function [SHR19].

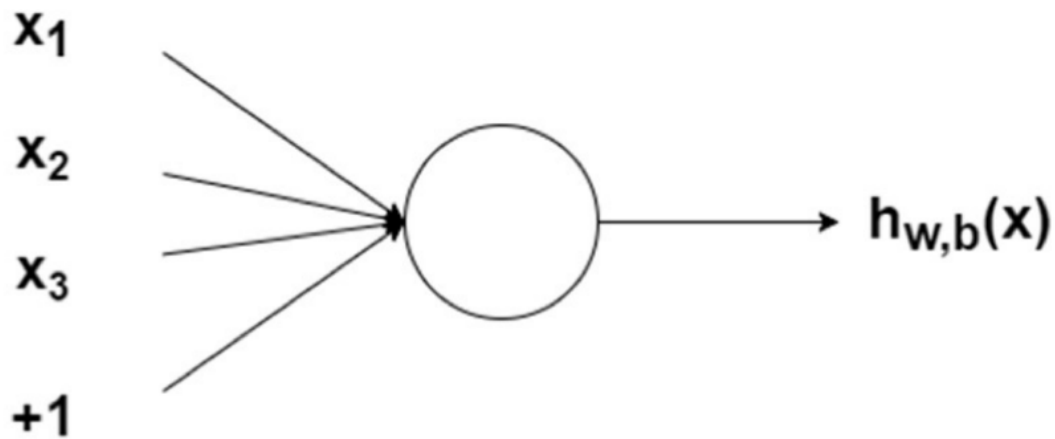
$$f(z) = \frac{1}{1 + e^{-z}}$$

In this equation, z is the input value and $f(z)$ is the activation function of the neuron. Each neuron has a node and a set of weighted inputs, along with a bias. The bias is an added value to the node that changes the overall output of the activation function.

When considering the bias b , the input to the node can be expressed as

$$n_{input} = x_1w_1 + x_2w_2 + x_3w_3 + b$$

Figure 2.5: Node with inputs, From [Pre16]



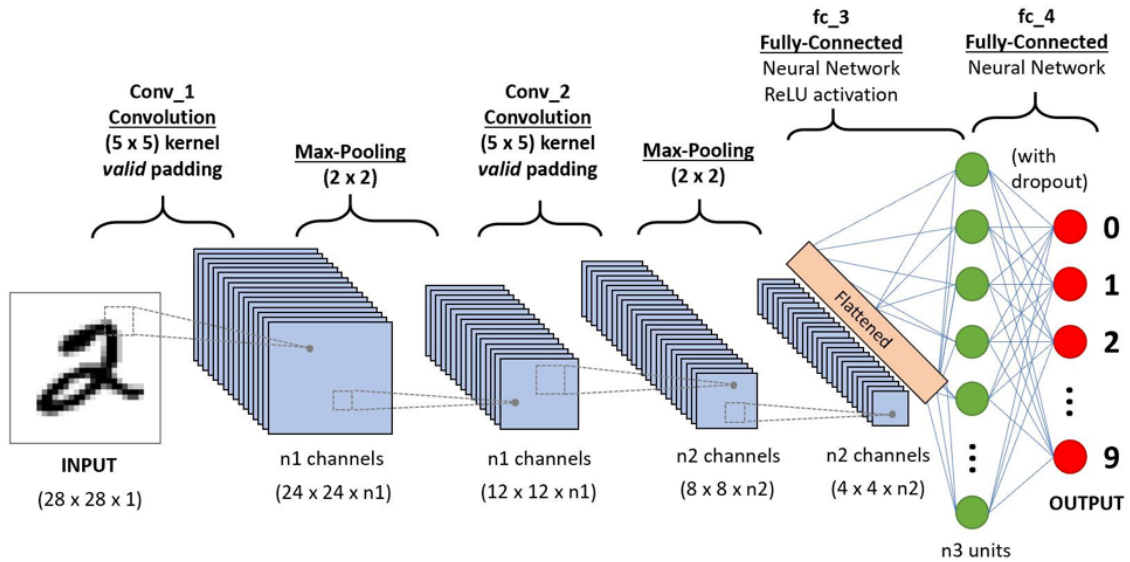
Where W_1 , W_2 and W_3 are the weights of the inputs. Neural networks come in different forms and functions. Nonetheless, we focus on three main neural networks that have proved their efficiency when generating abstractive summaries. Those neural networks are Convolutional Neural Networks and Recurrent Neural Networks.

2.5.2 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a special Artificial Neural Network architecture within which convolutional analysis is the basis of understanding of the features. Put in perspective, most of the features in the simpler form of neural network are analyzed as vectors and matrices, but the values of nearby features do not affect the analyzed component.

Contrarily to the simple Artificial Neural Network architecture, the input is analyzed as kernel and computed using the overlapping values between the kernel and the input [KGB14]. Most of the time, the input will be an image or any element that can be represented as a matrix. The hypothesis behind convolutional neural networks is that elements that are spatially close have influence over each other. This is highly true when

Figure 2.6: Convolutional Neural Network, From [Sum18]



processing images and in some instances, it also applies to text as textual elements have impact over other textual elements surrounding them [LB05]. If the image is denoted as f and the kernel as h , the indexes and rows of the image data denoted as m and n , we can determine the feature map values.

$$[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - k]$$

As presented in part 3.3 in reference to text coherence, the proximity of semantic elements of similar nature is the proof of coherence in the document. Now, the main issues that arises is how to represent a document in a valid form for interpretation by a CNN. CNN have been used successfully for operations such as sentiment analysis and text summarization. As explained in [CAR16], let d denote the dimension of word embedding, and s a document phrase consisting of a sequence of n words (w_1, w_2, \dots, w_n) which can be represented by the dense column matrix W belonging to the space $\mathbb{R}^{n \times d}$. Applying a

temporal narrow convolution between our input matrix W and the kernel K belonging to $\mathbb{R}^{c \times d}$ of width c , we obtain the following equation:

$$f_j^i = \tanh(W_{j:j+c-1} \otimes K + b)$$

Where \otimes is the Hadamard product followed by a sum over all elements. f_j^i is the feature j of the feature map f_i with bias b . Max Pooling is applied to get a single feature representing the phrase under the kernel K so that:

$$S_{i,K} = \max_j f_j^i$$

At the end, for each phrase, we obtain a phrase vectors that can later be used in other models for further analysis.

2.5.3 LSTM

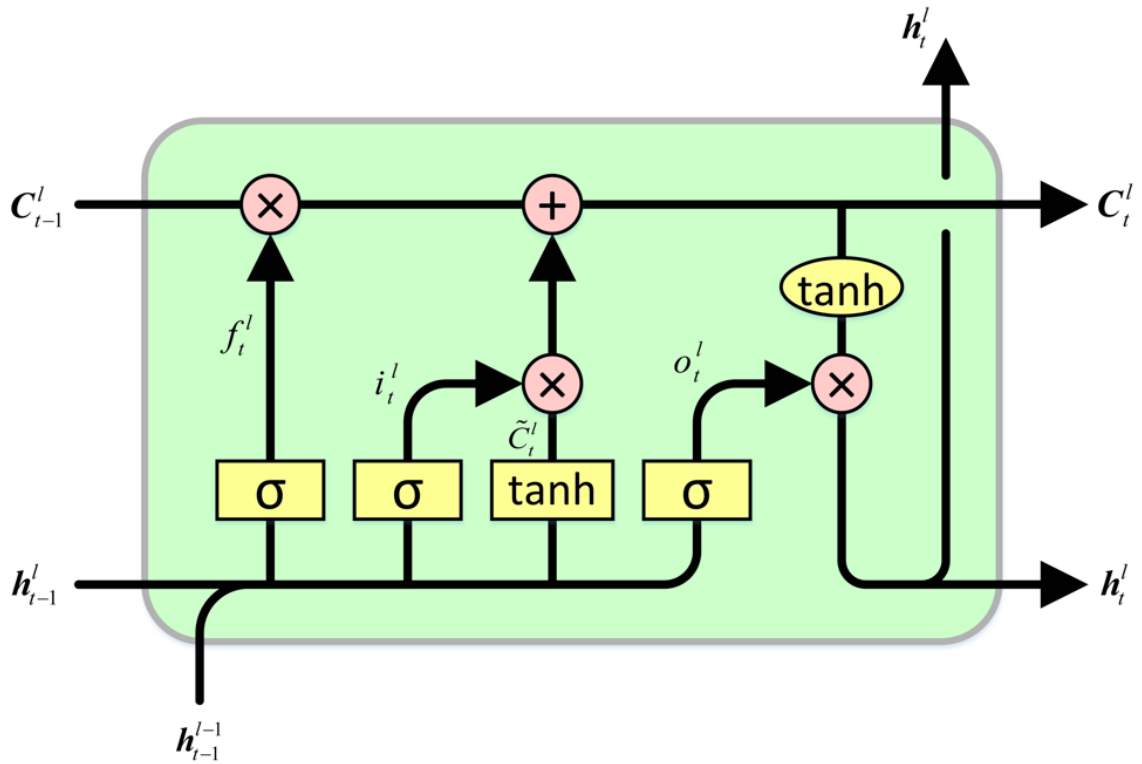
Long Short-Term Memory [LXZZ17] is a type of neural network widely used in problem such as speech recognition and language processing for its capacity to learn order dependence in sequence prediction problems. LSTM is a special type of RNN. A Recurrent Neural Network is multiple copies of a similar network, each providing a message to the next network in the chain. Recurrent Neural Networks provide a new way to analyze data based on time delay and information transfer. In our specific case, LSTM are the second layer of the CNN-LSTM [ON15] model we desire to use for our architecture.

The LSTM is made of 4 principal gates which are:

- The forget gate uses the output of the previous state $h(t-1)$ and takes decision about what needs to be deleted, thus keeping only important information. It is generally associated with a sigmoid function and is represented by the function

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Figure 2.7: Sample LSTM, From [dpr19]



- The input gate, which decides how much to add from the present input within a particular scale. It is defined by the functions

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- The present state is given by:

$$C_t = f_t * C_{t-1} + i_t * C_t$$

- The output gate that decides what will be the result by using a sigmoid function. It is given by the equation

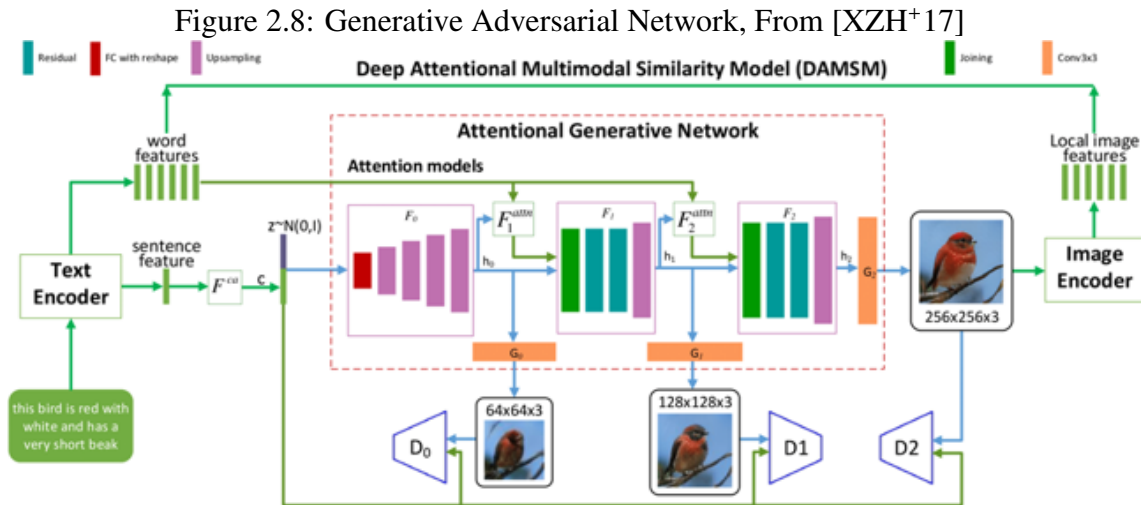
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

This system improves the result of the usage of recurrent neural networks by cancelling the gradient vanishing and reducing the time complexity of the training.

2.5.4 Generative Adversarial Network (GAN)

A GAN [Mer19] is a deep learning generative model. Is made of two essential components that are the Generator, that creates new examples for the problem, and the Discriminator, that is used to make the distinction between real and fake images.



The generative aspect of GANs and their capacity to produce relevant result out of information provided by the user makes them a great tool for generating frame sequences for our work. Given the discriminator D and the fixed generator G , the optimal D is given by the formula:

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

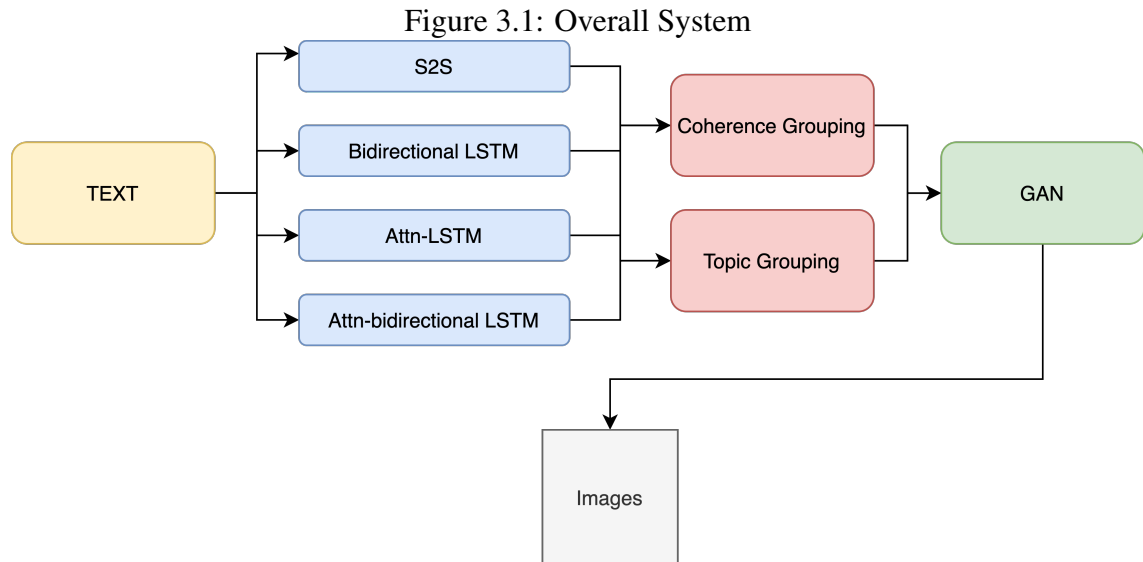
The discriminator is allowed to reached an optimum based on G and hence the convergence of P_g toward P_{data} .

3 CHAPTER III: METHODOLOGY

The ideology behind this experiment is to evaluate how the summarization and focus on the text coherence can impact the end result of a generative procedure that transforms text to image using GANs. We take advantage of the methodologies and process already present to help us fine tune our summarization procedures and create seeds that will perform better in term of text to image conversion. To initiate the research and ensure the success of our methodology, we first apply state of the art cleaning techniques and on our text data. This increases the precision of our algorithms and enhances the scoring applied to the sentences post pre-processing. The next step concerns the summarization that will be operated using three techniques well known and assessing the impact of coherence boosting to the result of the summarization using Rouge score. This is important to prove that an increase in the summarization quality can be made by focusing on the coherence but similarly, that the coherence of the summaries can be increased through coherence based scoring. The grouping of the sentences in order of topic relatedness is the next step of our operation and we expect the average seed coherence to be maximized. Finally, our formatted seeds will be converted into images along with non-grouped summaries to establish a qualitative analysis of the resulting images. The final group of images provided as output of the GANs will not be evaluated based on a metrics as no image quality metrics is provided to this day. Nevertheless, the capacity for the image to appear usable as an illustration is one of the fundamental criteria of the success of this operation.

3.1 Dataset

The main goal of this paper is to prove the capacity of the solution propose to convert complex text in a graphical representation that can be interpreted by a human without difficulty. To do so, we decided to use the corpus India News Summaries that has stories about actions, elements and persons, and would be great to express our capacity to create



content out of relevant text. As explained earlier, the very first phase of our analysis will be to prepare our text for the summarization algorithm by cleanup and tokenization.

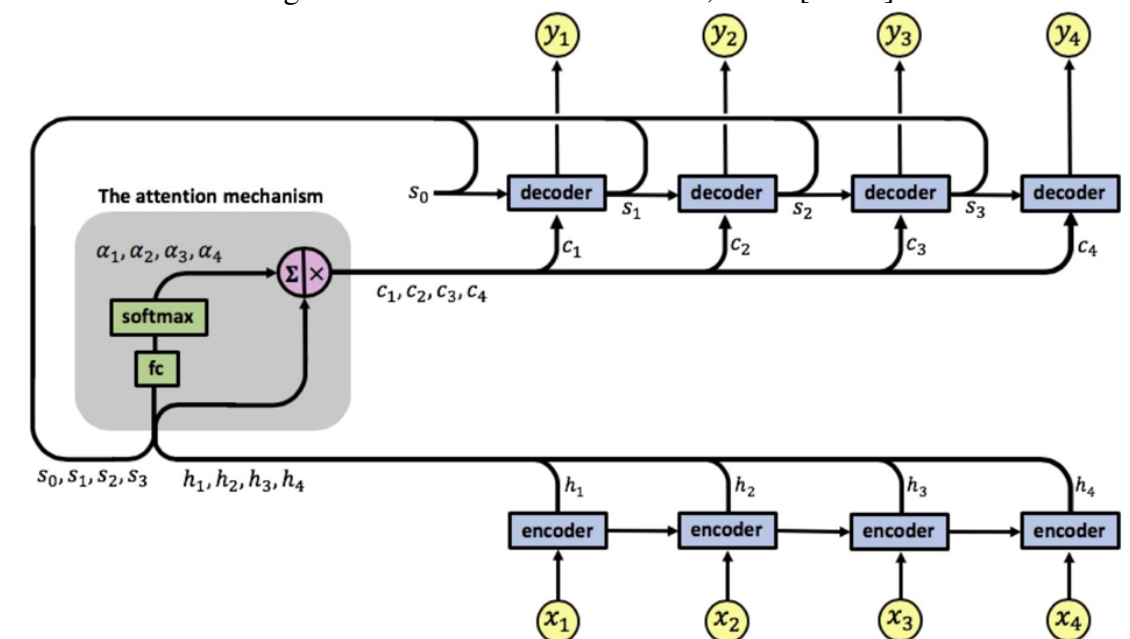
3.2 Models

To achieve a clear and objective summarization, we approach it using four different models, selected for their relative efficiency in the definition of summaries and their reliability. These three models are Attention-base LSTM, Seq2Seq, Bidirectional LSTM and a self-implemented Attention-based bidirectional LSTM. The later model did not have any previous reference and will be used to measure the impact of Attention layer on the overall result. We also used a simple version of TextRank to see the difference in results between abstractive summarization and extractive summarization.

3.2.1 Attention-Based LSTM

As explained in Fig 3.2, The attention based LSTM adds an attention layer [BCB14, Mer19]. The relative problem with encoder decoder is being able to compress the necessary information into a source vector, hence making it difficult to cope with long sentences. This

Figure 3.2: Attention Based LSTM, From [Nir19]

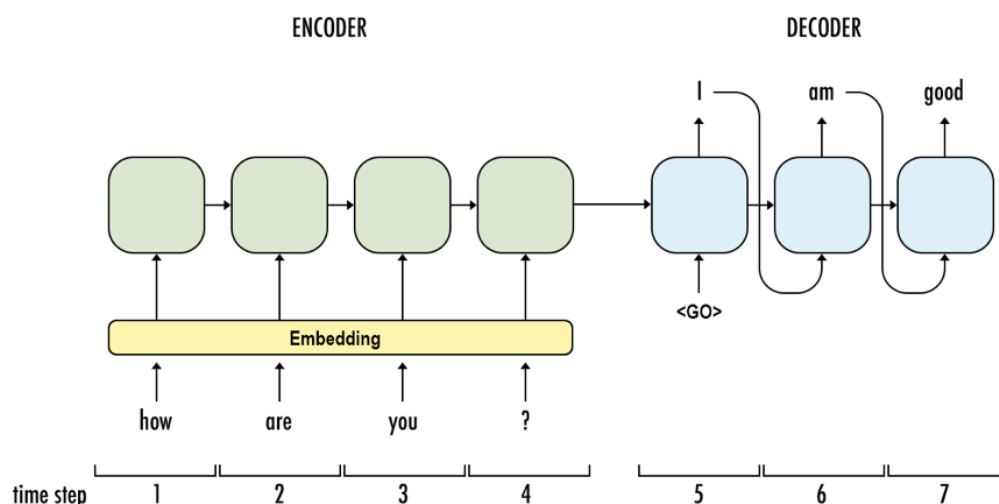


is why attention layer helps lift the weight of encoding sequences into fixed length vectors from the encoder by defining the input sequence into a set of vectors and choosing a subset of these vectors adaptively during the translation or summarization [BCB14].

3.2.2 Seq2Seq

Seq2Seq is the base model used in any form of summarization or translation. It uses an encoder and a decoder with fixed length input vectors to determine which word or sentence can be used as a result of a specific input. As you can see in Fig 3.3. It uses an embedding layer that feeds into an LSTM or RNN (Encoder) , and this one provides step information for another LSTM (decoder) to provide translation, summary or response.

Figure 3.3: Seq2Seq, From [Cha19]



3.2.3 Bidirectional LSTM

Bidirectional LSTM runs input in two ways, one from the past, and one from the future. This model preserves information from the future and using the two hidden states combined, we are able to preserve both present and future.

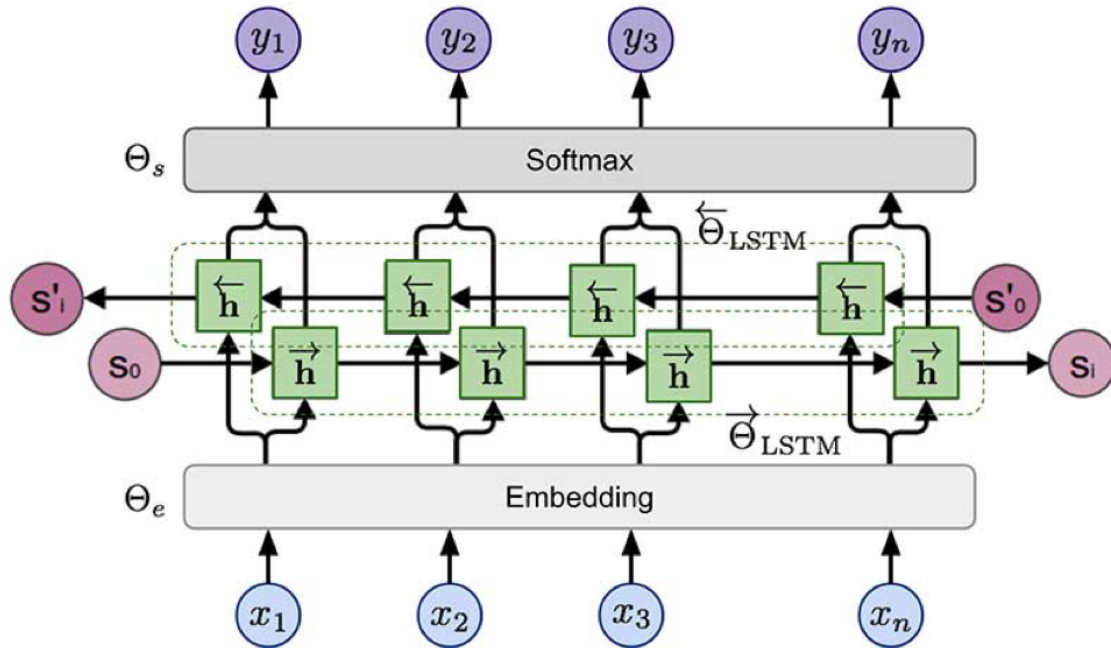
3.2.4 Attention-based Bidirectional LSTM

Attention-based Bidirectional LSTM [ZST⁺16] pushes further the capacity of the bidirectional LSTM by adding an Attentional layer in between the encoder and the decoder, already using a bidirectional LSTM. The advantages provided in the definition of the Attention-based LSTM are the same, providing a slight boost over the already well performing bidirectional LSTM.

3.2.5 TextRank

TextRank is a widely used and approved extractive summarization model used to identify the most important words or sentences in a text. This model is used as a benchmark

Figure 3.4: Bidirectional LSTM, From [Lil19]



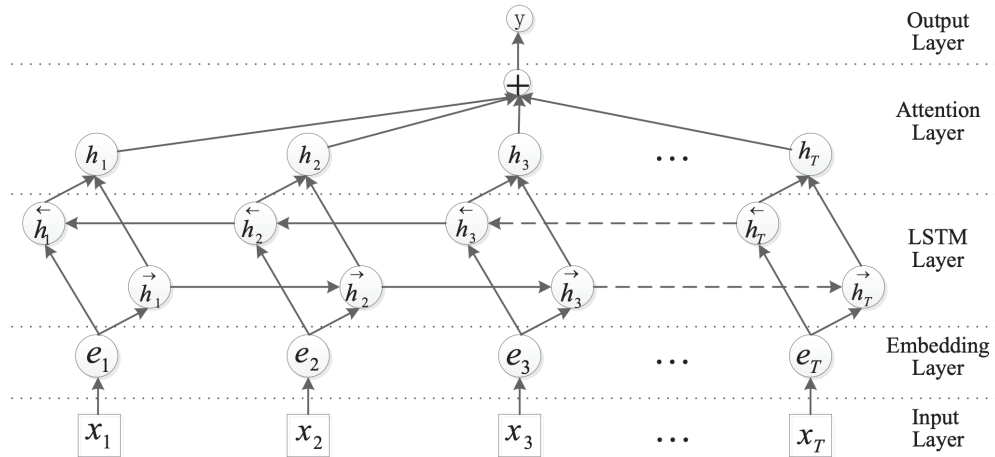
to appreciate the performance of the other models and also, provide a starting point to the conversion of text to images.

3.2.6 Attentional Generative Adversarial Network (Attn-GAN)

AttnGan is a Generative Adversarial Network that uses an Attention layer and enables the conversion of text to images. Our goal in our experiment is not to improve the model but use it as a tool to evaluate the impact of different summarization techniques on the end result image.

3.3 Preprocessing

To preprocess our data, we went through intense data cleaning, during which every non desirable character was removed from texts and summaries. We also applied a lemmatization and removed words that appear more than 100 times in the whole document. We then proceeded with a tokenization of the dataset, using 65 characters as the maximum

Figure 3.5: Attention based Bidirectional LSTM, From [ZST⁺16]

number of characters required for the text and 15 the maximum number of characters required for the summary. We later proceeded to the implementation of the different models and set our operations.

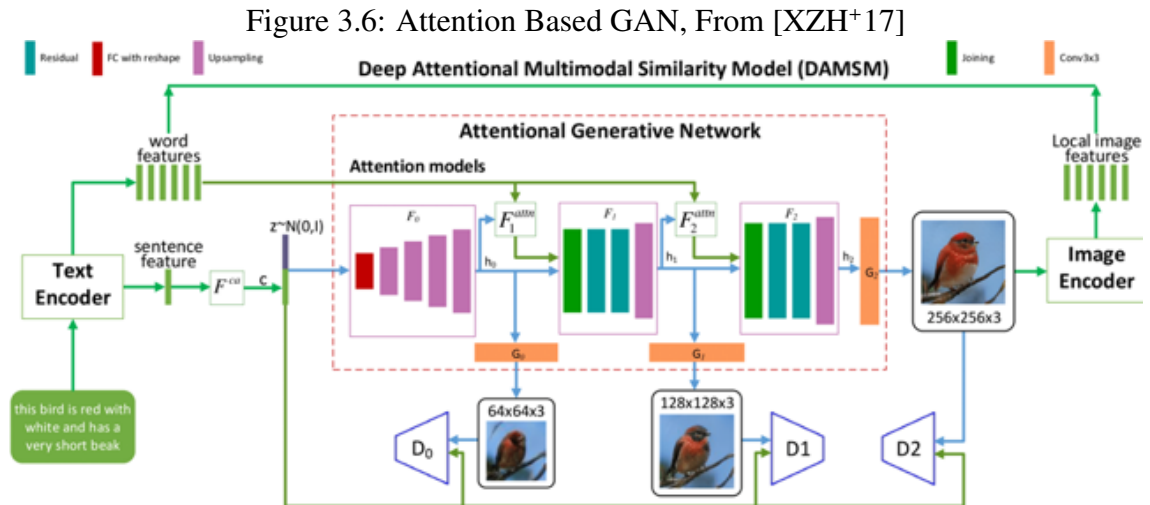
3.4 Environment

The achievement of this research was possible with the use of a single device. The size of the dataset (<200MB) and the low complexity of the models and lack of need for supplementary training made it easy to run the operations on a single machine.

The hardware used was a mid 2015 MacBook Pro with 2.7 GHz Intel Core i5, one graphic card Intel Iris Graphics 6100 1536 MB, 8 GB 1867 MHz DDR3 of RAM storage and 256 GB SSD. The storage was the main issue at the beginning of the experiment when using larger dataset. But the reduction in size helped permit the experiment to hold true.

The software used was the following:

- OSX High Sierra : Operating System hosted on the Mac pro.



- Jupyter Notebook: Integrated development environment used to build and visualize the models.
- Pycharm: Integrated development environment used to preprocess the data.
- Python 3.7: Programming language used for the research
- Keras-CPU: Machine learning Framework with Tensorflow backend. Useful to build neural network and tune hyperparameters.
- OpenCV: Computer vision framework used for preprocessing.
- Tensorflow 3: Machine Learning framework provided by Google

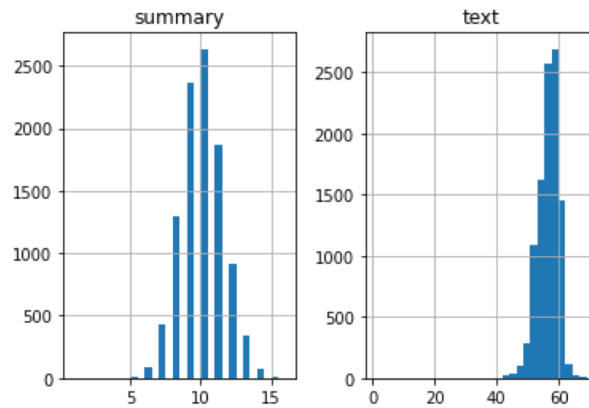
Figure 3.7: Example of generated image, From [XZH⁺17]

4 CHAPTER IV: RESULT

4.1 Data Understanding

The very first step to understanding our results sits in the definition of our dataset and the preliminary operations. As we identified before, we decided to go with 65 characters as base input and 15 characters limit for the summary based on the training data.

Figure 4.1: Summaries and articles lengths histogram



4.2 Models Setup

The different models used for this experiment were Seq2Seq, Attn-LSTM, Bidirectional LSTM and Attn-Bidirectional LSTM. Each model was trained with a total of 10000 data points and a vocabulary of 7994 unique words. The latent dimension for each LSTM layer was 300 and we set our embedding layer to have a dimension of 200. The learning rate was set to $2.5 * 10^{-5}$, We also set an early stopping with a patience of 5 based on the validation loss. All these elements can be found in the following table.

| Parameter | Value |
|---------------------------|----------|
| Dataset | 10000 |
| Vocabulary | 5394 |
| LSTM Layer Dimension | 500 |
| Embedding Layer Dimension | 150 |
| Learning Rate | 0.000025 |
| Early Stopping patience | 5 |

Table 4.1: Hyper-Parameters

4.3 Models Results

4.3.1 Models Validation and Training Loss curves

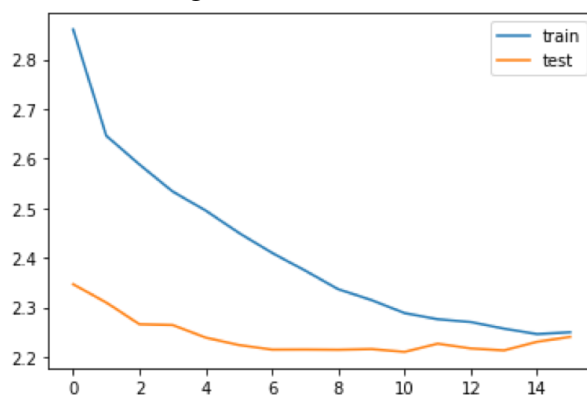
In Fig 4.2, We clearly observe that the training of the model provided sub-par results and ended in a training loss of 2.275 and a validation loss of 2.258. This can be due to a learning rate too high for the complexity of the model or a dataset that is not clean enough.

For our three other models in Fig 4.3, Fig 4.4 and Fig 4.5, We observe that the training loss has a continuously decreasing curve and the validation loss has a curve that saddles around 2.087. These values provided medium level summaries whose Rouge-Score evaluation provided the following result table.

| | model | Rouge1 | Rouge2 | RougeL |
|---|-------------------------|----------|----------|----------|
| 0 | Attention-LSTM | 0.082377 | 0.008868 | 0.105263 |
| 1 | S2S | 0.157929 | 0.025037 | 0.153168 |
| 2 | Bidirectional-LSTM | 0.289683 | 0.083734 | 0.288116 |
| 3 | Attn-Bidirectional LSTM | 0.329866 | 0.107328 | 0.33508 |

Table 4.2: Rouge scores for our different summarization models

Figure 4.2: Training and validation loss for Attn-LSTM



The rouge scores present the clear superiority of the Attn-bidirectional LSTM model in term of abstractive text summarization compared to the other models under an evaluation based on the same parameters.

4.3.2 Coherence results and LDA results

The coherence score of the summaries, grouped by set of three with summary coherence grouping threshold:

We observe in figure 4.6 that the grouped summaries coherence is slightly higher, compared to the coherence provided during the analysis of the complete articles as you can see in figure 4.7

The association of summaries by topic did not have a high impact on the end result as most of the individual summaries diverge in topic.

4.3.3 GAN Images

The different sample images were generated using the Attn-Bidirectional LSTM summarization model and coherence grouping.

Figure 4.3: Training and validation loss for Seq2Seq

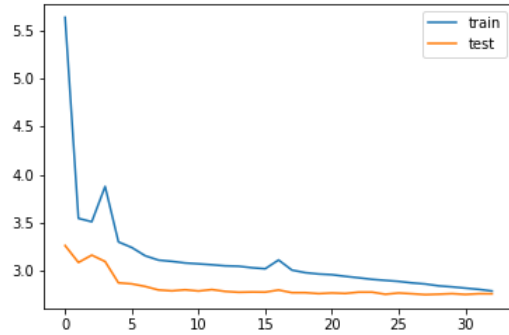


Figure 4.4: Training and validation loss for Bidirectional LSTM

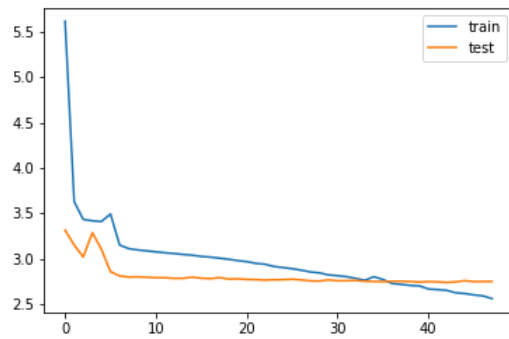


Figure 4.5: Training and validation loss for Attn-Bidirectional LSTM

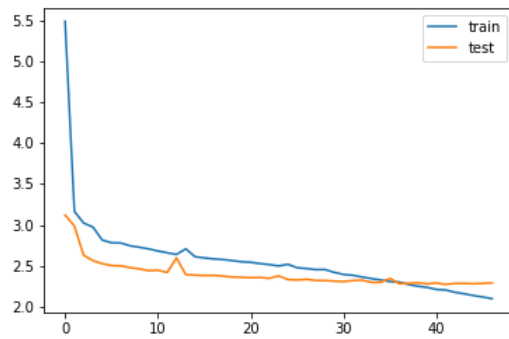


Figure 4.6: Summaries Coherence histogram

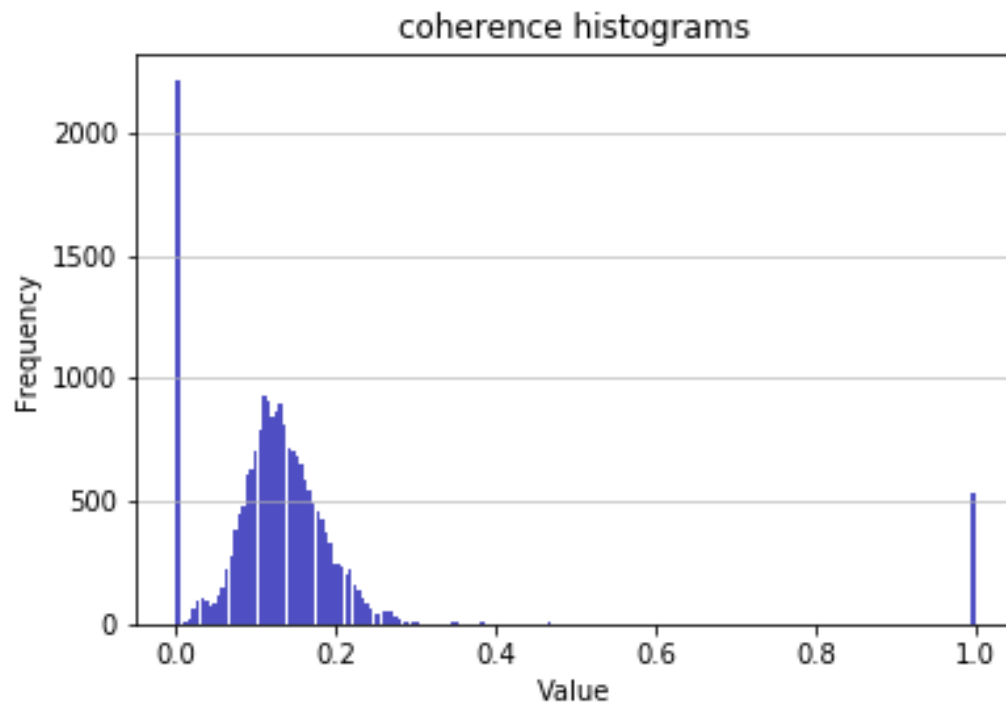


Figure 4.7: Articles Coherence histogram

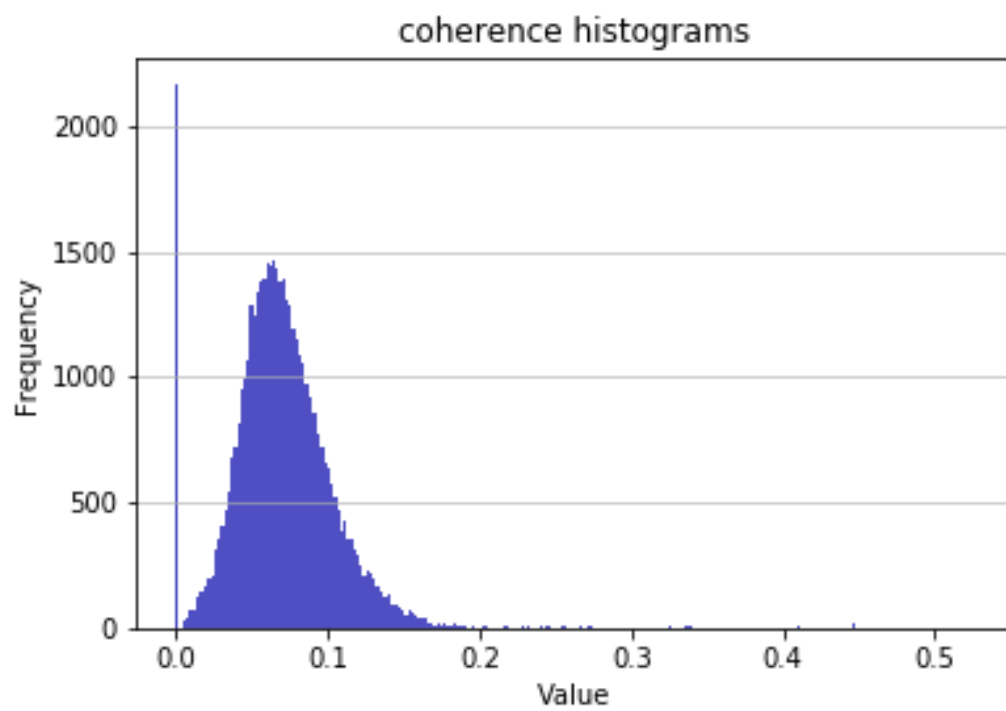
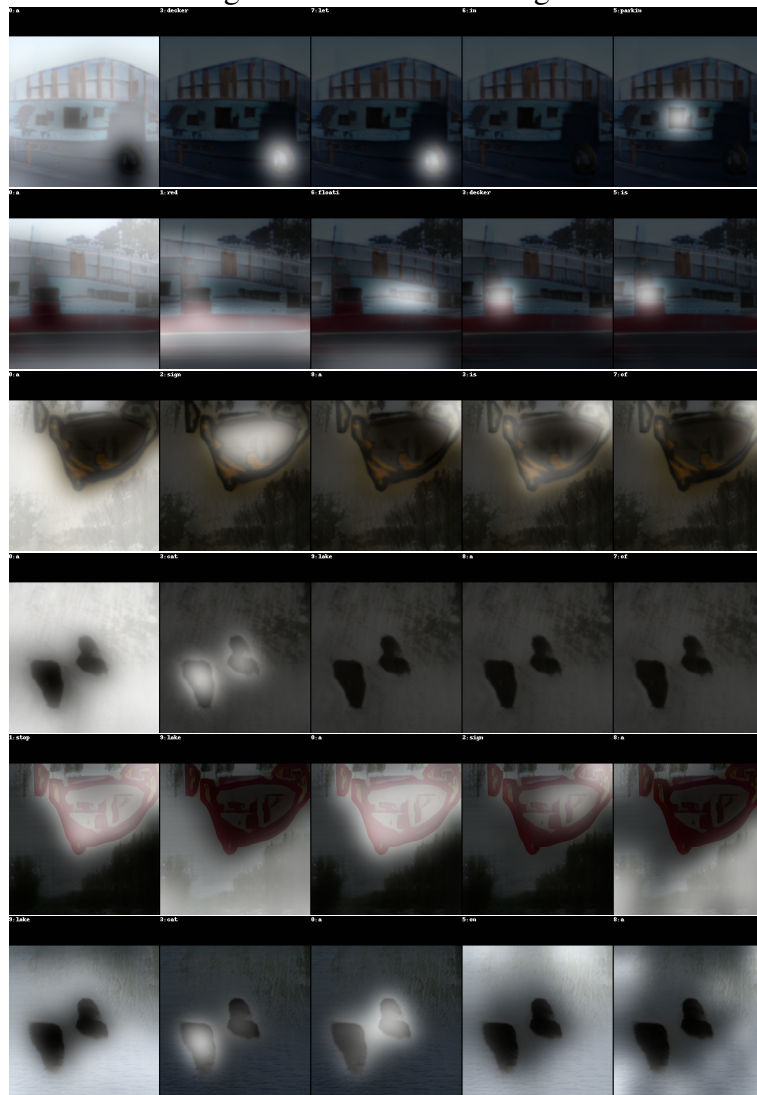


Figure 4.8: Generated images



5 CHAPTER V: CONCLUSION

Information extraction is a branch of text mining that discusses the means and methods to gather useful information from a text. The generation of images from text is a novel topic that is progressively gaining traction and changing the way we approach text representation. Our goal in this research was to assess the pipeline of transformation of long text in set of images and provide a qualitative assessment over the images generated. We observed that while the quality of the summarization has a direct impact on the quality of the provided image at the end of the operation, elements such as the coherence or topic grouping have a minimal impact on the overarching result. In fact, the coherence grouping only helps in situations where there is not enough information in one summary to produce a valuable image. While the qualitative assessment of the image was used as judgement, a desire for a methodology to grade the realism of the result image, independently of the dataset used, is really important and would shine the light on new possibilities in term of text processing and image processing.

REFERENCES

- [AA09] Rasim ALGULIEV and Ramiz ALIGULIYEV. Evolutionary Algorithm for Extractive Text Summarization. *Intelligent Information Management*, 01(02):128–138, 2009. doi:10.4236/iim.2009.12019
- [Bar73] Jonathan Baron. Semantic components and conceptual development. *Cognition*, 2(3):299–317, 1973. doi:10.1016/0010-0277(72)90037-6
- [BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL: <http://arxiv.org/abs/1409.0473>
- [CAR16] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 93–98, 2016. doi:10.18653/v1/n16-1012
- [CCP19] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. pages 40–46, 2019. doi:10.18653/v1/w18-5406
- [Cha19] Chaoran. Build a machine translator using keras (part-1) seq2seq with lstm, 2019. [Online; accessed June, 2019]. URL: <https://raw.githubusercontent.com/6chaoran/nlp/master/nmt/image/encoder-decoder-architecture.png>
- [Cla13] Stephen Clark. Topic Modelling and Latent Dirichlet Allocation. 2013.
- [CY04] Lin Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. *Japanese Circulation Journal*, 34(12):1213–1220, 2004. doi:10.1253/jcj.34.1213
- [dpr19] dprogrammer. Rnn, lstm gru, 2019. [Online; accessed November, 2019]. URL: https://cdn.shortpixel.ai/client/q_glossy,ret_img,w_768/http://dprogrammer.org/wp-content/uploads/2019/04/LSTM-Core-768x466.png
- [FDL⁺13] Rafael Ferreira, Luciano De Souza Cabral, Rafael Dueire Lins, Gabriel Pereira E Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima, Steven J. Simske, and Luciano Favaro. Assessing sentence scoring techniques for extractive text summarization. *Expert Systems with Applications*, 40(14):5755–5764, 2013. doi:10.1016/j.eswa.2013.04.023
- [FvKS04] U Flick, E von Kardoff, and I Steinke. Mayring, P., 2004. Qualitative content analysis. *A Companion to Qualitative Research*, 2004. URL: <https://books.google.de/books?id=1RSL1KJjEPoC>

- [Gey07] Charles J Geyer. Stat 5102 Notes: Brand Name Distributions. *Brand*, pages 1–20, 2007.
- [HC16] Markus Hofmann and Andrew Chisholm. *Text Mining and Visualization: Case Studies Using Open-Source Tools*. 2016.
- [HF95] John R. Hughes and Tonya Fuller. The Use of Felbamate in Patients with Periodic Lateralized Epileptiform Discharges (PLEDs). *Clinical EEG and Neuroscience*, 26(3):180–183, 1995. doi:10.1177/155005949502600310
- [KB18] P. Krishnaveni and S. R. Balasundaram. Automatic text summarization by local scoring and ranking for improving coherence. In *Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017*, volume 2018-Janua, pages 59–64, 2018. doi:10.1109/ICCMC.2017.8282539
- [KGB14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1:655–665, 2014. arXiv:1404.2188, doi:10.3115/v1/p14-1062
- [Kuz15] O. P. Kuznetsov. Conceptual semantics. *Scientific and Technical Information Processing*, 42(5):307–312, 2015. doi:10.3103/S0147688215050044
- [Lan06] Thomas K Landauer. Latent Semantic Analysis. *Encyclopedia of Cognitive Science*, pages 1–14, 2006. doi:10.1002/0470018860.s00561
- [LB05] Mirella Lapata and Regina Barzilay. Automatic evaluation of text coherence: Models and representations. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1085–1090, 2005.
- [LH13] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8206 LNCS:611–618, 2013. doi:10.1007/978-3-642-41278-3_74
- [Lil19] Lilian, Weng. Generalized language models: Cove, elmo cross-view training, 2019. [Online; accessed March 27, 2020]. URL: https://www.topbots.com/wp-content/uploads/2019/04/cover_ELMo_web-1280x640.jpg
- [Lin16] Jiayu Lin. On The Dirichlet Distribution. (September), 2016. URL: <https://mast.queensu.ca/~communications/Papers/msc-jiayu-lin.pdf>0Ahttp://mast.queensu.ca/~communications/Papers/msc-jiayu-lin.pdf
- [LP91] Beth Levin and Steven Pinker. Introduction to special issue of Cognition on lexical and conceptual semantics. *Cognition*, 41(1-3):1–7, 1991. doi:10.1016/0010-0277(91)90030-8

- [LXZZ17] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. *Advances in Neural Information Processing Systems*, 2017-Decem:4089–4099, 2017.
- [Man09] Christopher D. Manning. Intro to Information Retrieval. *Information Retrieval*, (c):1–18, 2009. arXiv:0521865719 9780521865715, doi:10.1109/LPT.2009.2020494
- [Mer19] Stephen Merity. Single Headed Attention RNN: Stop Thinking With Your Head. 2019. URL: <http://arxiv.org/abs/1911.11423>, arXiv:1911.11423
- [MH91] J MORRIS and G HIRST. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17(1):21–48, 1991.
- [Moh13] Sabah Mohammed. Introducing the new JETWI associate editor-in-chief. *Journal of Emerging Technologies in Web Intelligence*, 5(1):1, 2013. doi:10.4304/jetwi.5.1.1-1
- [MS12] Suneetha Manne and S. Sameen Fatima. A Feature Terms based Method for Improving Text Summarization with Supervised POS Tagging. *International Journal of Computer Applications*, 47(23):7–14, 2012. doi:10.5120/7494-0541
- [Nir19] Nir,Arbel. Attention in rnns, 2019. [Online; accessed December, 2019]. URL: https://miro.medium.com/max/1400/1*wnXVyE8LXPfODvB_Z5vu8A.jpeg
- [ON15] Keiron O’Shea and Ryan Nash. An Introduction to Convolutional Neural Networks. pages 1–11, 2015. URL: <http://arxiv.org/abs/1511.08458>, arXiv:1511.08458
- [Pre16] Preeti,Verma. In a nutshell: Neural networks, 2016. [Online; accessed November, 2019]. URL: <https://thebeautyofml.files.wordpress.com/2016/03/singleneuron.png?w=768>
- [QSF+18] Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? pages 529–535, 2018. doi:10.18653/v1/n18-2084
- [Sar07] S Sarawagi. Information Extraction. *Communications of the ACM*, 1(3):261–377, 2007. URL: <http://dl.acm.org/citation.cfm?id=234209>, doi:10.1561/1500000003
- [SB19] José Saura and Dag Bennett. A three-stage method for data text mining: Using ugc in business intelligence analysis. *Symmetry*, 11:519, 04 2019. doi:10.3390/sym11040519

- [SHR19] Shengli Song, Haitao Huang, and Tongxiao Ruan. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, 78(1):857–875, 2019. doi:10.1007/s11042-018-5749-3
- [Sum18] Sumit,Saha. A comprehensive guide to convolutional neural networks — the eli5 way, 2018. [Online; accessed December, 2018]. URL: https://miro.medium.com/max/1400/1*uAeANQIOQPqWZnnuH-VEyw.jpeg
- [WB11] Chong Wang and David M. Blei. CTM for Recommending Scientific Articles. *Kdd*, 2011.
- [WK92] Jonathan Webster J. and Chunyu Kit. Tokenization as the Initial Phase in NLP. *Procession of COLING-92*, (27):p517–522, 1992.
- [XZH⁺17] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *CoRR*, abs/1711.10485, 2017. URL: <http://arxiv.org/abs/1711.10485>, arXiv:1711.10485
- [Yil12] Ilker Yildirim. Bayesian Inference: Gibbs Sampling. *Department of Brain and Cognitive Sciences*, 14627:1–6, 2012. URL: <http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf>
- [ZST⁺16] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, pages 207–212, 2016.