People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Yahia Fares - Medea

Faculty of Science

Department of Mathematics and Computer Science



Graduation Project

Submitted for Graduation in Computer Science Master

Specialty: *System Engineering and Web Technology*

Novel Attention Deep Learning Model for Cross Language Abstractive Text Summarization

Supervised by

Dr. Hentabli Hamza

Submitted by

Ms. Berrabia Nabila

Mr. Fekhari Abdelouahab

2021/2022

ACKNOWLEDGEMENT

We thank God, the Almighty, for giving us the strength to follow our studies as well as the courage to overcome all the difficulties.

We would like to sincerely thank Dr. HENTABLI Hamza, who has always been attentive and very available throughout the production of this project.

The jurors for their time and attention to our work. teachers from our university, the Mathematics Department, and Computer Science.

Finally, we send our sincere thanks to all of our relatives and friends, who have always supported and encouraged us during the production of this project.

DEDICATION

We take this opportunity to express our gratitude our guide, *Dr:Benaissa Azzeddine Rachid* for his exemplary guidance, monitoring, and constant encouragement throughout the course of this project

We take this opportunity to express a deep sense of gratitude to our friends for their support and motivation which helped us in completing this task through its various stages.

We would like to thank our parents, who have never failed to give moral support, for giving us all our needs during the time we developed our project, and for teaching us that even the largest task can be accomplished if it is done one step at a time.

Finally, we dedicate this project to all the people who have worked hard to help us complete this project.

Berrabia Nabila and Fakhari Abdelouahab

ABSTRACT

In the age of information, the huge amount of data that is produced every day will remain unuseful for humans unless we make it available with new tools and technologies. Abstractive Text Summarization tries to get the most essential content of a text corpus and compress it into a shorter text while keeping its meaning and maintaining its semantic and grammatical correctness. Neural architectures are becoming dominant in Abstractive Text Summarization. The use of deep learning architectures in natural language processing entered a new area after the appearance of seq2sep models in recent decades. These models are basically founded on a couple of recurrent neural networks that connect the input and output data in an encoder-decoder architecture. Better results were possible by adding the Attention Mechanism to the RNN layers. Many studies have shown how competitive these new architectures are in machine translation. The recently proposed model, which is called Abstractive Text Summarization (ATS), uses only the attention mechanism. It has surpassed the previous state-of-the-art models in translation tasks. In this work, we want to experiment with this new model for abstractive text summarization. The evaluation results on the *Amazon fine food reviews* dataset show that our method better captures the attention pattern and achieves The Bleu scored equivalent (0,50). The result is a very high-quality summary. Abstractive Cross-Language Summarization CLA (translation and summarization) is the task of generating a summary in a target language from a collection of documents in a different source language. The result of the blue score was very good and satisfying (0.50).

Résumé

À l'ère de l'information, l'énorme quantité de données produites chaque jour restera inutile pour les humains à moins que nous ne les rendions disponibles avec de nouveaux outils et technologies. Abstractive Text Summarization essaie d'obtenir le contenu le plus essentiel d'un corpus de texte et de le compresser en un texte plus court tout en gardant son sens et en maintenant son exactitude sémantique et grammaticale. Les architectures neuronales deviennent dominantes dans le résumé de texte abstrait. L'utilisation d'architectures d'apprentissage en profondeur dans le traitement du langage naturel est entrée dans un nouveau domaine après l'apparition des modèles seq2sep au cours des dernières décennies. Ces modèles reposent essentiellement sur quelques réseaux de neurones récurrents qui connectent les données d'entrée et de sortie dans une architecture encodeur-décodeur. De meilleurs résultats ont été possibles en ajoutant le mécanisme d'attention aux couches RNN. De nombreuses études ont montré à quel point ces nouvelles architectures sont compétitives en traduction automatique. Le modèle récemment proposé, appelé Abstractive Text Summarization (ATS), utilise uniquement le mécanisme d'attention. Il a dépassé les précédents modèles de pointe dans les tâches de traduction. Dans ce travail, nous voulons expérimenter ce nouveau modèle de résumé de texte abstrait. Les résultats de l'évaluation sur l'ensemble de données des critiques d'épicerie fine d'Amazon montrent que notre méthode capture mieux le modèle d'attention et atteint l'équivalent du score The Bleu (0,50). Le résultat est un résumé de très haute qualité. Résumé abstrait multilingue CLA (traduction et résumé) est la tâche de générer un résumé dans une langue cible à partir d'une collection de documents dans une langue source différente. Le résultat du score bleu était très bon et satisfaisant (0,50)

الـمـلـخـص

في عصر المعلومات، ستظل الكمية الهائلة من البيانات التي يتم إنتاجها يوميًا غير مفيدة للبشر ما لم نوفرها بأدوات وتقنيات جديدة. ينقسم التلخيص الأتوماتيكي إلى نوعين Extractiveو Abstractive . يستعمل التلخيص الأوتوماتيكي Abstractive لاستخراج أهم المعلومات الموجودة في النص وضغطها في نص أقصر، مع الحفاظ على معناها والحفاظ على دلالتها وصحتها النحوية. في السنوات الأخيرة كثر استعمال ما يسمى بالتعلم العميق (Deep Learning) في مجال معالجة النصوص (NLP) وخاصة بعد ظهور نموذج encoder-decoder. تتكون نماذج encoder-decoder من عدة طبقات من الشبكات العصبية (RNN)، و يفصل بين encoder و decoder طبقة تسمى ب Attention-Layer، و قد ركزت الدراسات على أهميت Attention-Layer ومدى تأثيرها في النتائج ، و خاصة في الترجمة الآلية (machine translation) كما تزايد استعمالها في مجال التلخيص(Abstractive). في هذه المذكرة و بالاستعانة ببيانات *Amazon fine food reviews*، قمنا بتجربة هذا النموذج RNN Encoder-Decoder with Attention Mechanism لبناء برنامج Abstractive Text Summarization ،وقد كانت النتائج اكثر من رائعة حيث كانت نسبة BLEU (0.50) ،كما قمنا بتجربته على Cross-Language Summarization CLA (الترجمة والتلخيص) حيث لغة الإدخال غير لغة الإخراج ، وكانت نتيجة الدرجة الزرقاء جيدة جدا ومرضية (0.50).

TABLE OF CONTENTS

ACKNOWLEDGEMENT ..... **II**

DEDICATION ..... **III**

ABSTRACT ..... **IV**

الملخص ..... **VII**

TABLE OF CONTENTS ..... **VIII**

LIST OF TABLES ..... **XII**

Introduction **1**

1.1 Introduction **1**

1.2 Problem Background **3**

1.3 Research Motivation **4**

1.4 Thesis Statement **4**

1.5 Decision Problems **5**

1.6 Thesis Objectives **6**

1.7 Thesis Scope **7**

Automatic text summarization **9**

2.1 Introduction **9**

2.2 Definition of Text Summarization **9**

2.3 The need for automatic summarization **10**

2.4 Related Works **10**

2.5 Characteristics of a summary **11**

2.6 Steps of automatic text summarization **12**

2.7 Concept of salient sentences and scoring sentence **12**

2.8 Various types of summaries **13**

2.8.1 Summary based on input 13

2.8.3 Summary based on content 14

2.8.4 Summary based on output 14
2.8.5 Summary based on language 15
2.9 Update summarization **16**
2.10 Text summarization methods **17**
2.11 Evaluation techniques **20**
2.11.1 Intrinsic evaluation 20
2.11.2 Extrinsic evaluation 22
2.12 Summarization Main Problems **23**
2.13 Conclusion **24**
Methodogy **25**
3.1 Introduction **25**
3.2 Research Design **25**
3.3 General Research Framework **25**
3.4 Dataset **28**
3.5 Recurrent neural networks (RNN) for text summarization **30**
3.6 Long short term memory (LSTM) : **32**
3.7 Learning Process of Deep Neural Networks **33**
3.7.1 Activation Functions: 34
3.7.2 Loss Functions 35
3.7.3. Gradient Descent 36
3.8 Machine Translation and Summarization **37**
3.9 Attention Layer **37**
3.9.1. Attention Mechanism 38
3.9.2. Attention in Translation 41
3.9.3. Implementation of Attention 43
3.10 Embeddings **44**
3.10.1 Word Embeddings 44
3.10.2 Sentence Embeddings 45
3.11 Dense layer **46**
3.12 Proposed Approach Steps **47**

3.12.1 Pre-processing 47
3.12.2 Attention in abstractive summarization 49
3.12.3 Sequence-to-Sequence (Seq2Seq) Modeling 50
3.12.4 Prediction 52
3.13 Architecture **53**
3.13.1 Encoder 53
3.13.2 Decoder 54
3.14 Abstractive cross-language summarization (Translate and Summarize) **54**
3.15 Conclusion **56**
Test and Evaluation **57**
4.1 Introduction **57**
4.2 Environment and Development Tools **57**
4.2.1 Material 57
4.2.2 Python Modules Used 57
4.4 Experimental Results and Discussion **60**
4.4.1 Results from ATS English-English 61
4.4.2 Results from ATS English-French 61
4.4.3 Results from ATS English-Arabic 62
4.4.4 Results from ATS French-French 62
4.4.5 Results from ATS French-Arabic 63
4.4.6 Results from ATS French-English 63
4.4.7 Results from ATS Arabic-Arabic 64
4.4.8 Results from ATS Arabic-English 64
4.4.10 Results from ATS Train Arabic-Test French 65
4.4.11 Results from ATS Train Arabic-Test English 66
4.4.12 Results from ATS Train French- Test English 66
4.4.13 Results from ATS Train French- Test Arabic 67
4.4.14 Results from ATS Train English-Test French 67
4.4.15 Results from ATS Train English-Test Arabic 68
4.5 The results of the Bleu and Rouge scores **68**
4.5.1 The results of the Bleu score 68
4.5.2 Sentence BLEU score for validation dataset 69
4.6 Answers to Research Questions **76**

4.6.1 Answer 1 76
4.6.2 Answer 2 77
4.6.3 Answer 3 77
4.6.4 Answer 4 77
4.7 Conclusion **77**
General Conclusion and perspective **78**
REFERENCES **79**

LIST OF TABLES

Table 2.1: Comparison between different Automatic Text Summarization approaches that exist according to their advantages and disadvantages. 19

Table 4.1: Results of the Bleu score. 68
Table 4.2: Results of the Bleu score (different input). 69
Table 4.3: Results of the Rouge score. 76

LIST OF FIGURES

Figure 2.1 : Extractive text summarization. 15
Figure 2.2: Abstractive text summarization. 15
Figure 2.3: Text summarization classification. 16
Figure 2.4: Evaluation techniques of Automatic Text Summarization. 23
Figure 2.5: Main steps of summarization. 24

Figure 3.1: Overall research framework and activities implemented in the thesis. *26*
Figure 3.2: Reviews dataset. *28*
Figure 3. 3: Reviews dataset (translated to Arabic). *29*
Figure 3.4: Reviews dataset (translated to French). *29*
Figure 3. 5: Reviews dataset (translated to Arabic, French). *30*
Figure 3.6: RNN Architecture. *31*
Figure 3.7: Bidirectional RNN Architecture with forward and backward hidden RNN-layers. *32*
Figure 3.8: Basic Architecture of LSTM. *33*
Figure 3.9: Illustration of deep neural network learning process (TORRES.AI, 2018). *33*
Figure 3.10: Deep learning model. *38*
Figure 3.11: Attention mechanism illustration. *39*
Figure 3.12: Attention mechanism illustration. *40*
Figure 3.13: Attention mechanism illustration. *41*
Figure 3.14: Example of language translation using LSTM. *42*
Figure 3.15: Example of language translation using LSTM and attention. *44*
Figure 3. 16: Sentence embedding illustration. *46*
Figure 3.17: Illustration of Proposed Abstractive summarization Framework. *47*
Figure 3.18: Pre-processing procedure illustration. *48*
Figure 3.19: Abstractive summarization preprocessing. *49*
Figure 3.20: Abstractive summarization attention implementation. *50*
Figure 3.21: Sequence-to-sequence model with attention mechanism. *51*
Figure 3.22: Abstractive summarization prediction illustration. *52*
Figure 3.23: Encoder-Decoder Sequence-to-Sequence. *53*
Figure 3.24: Encoder-Decoder. *54*

Figure 3.25: Abstractive summarization (English-Arabic). *55*
Figure 3.26: Abstractive summarization (Arabic-French). *55*

Figure 4.1: Results from ATS English-English. *61*
Figure 4.2: Results from ATS English-French. *62*
Figure 4.3: Results from ATS English-Arabic. *62*
Figure 4. 4: Results from ATS French-French. *62*
Figure 4.5: Results from ATS French-Arabic. *63*
Figure 4.6: Results from ATS French-English. *63*
Figure 4.7: Results from ATS Arabic-Arabic. *64*
Figure 4.8: Results from ATS Arabic-English. *64*
Figure 4.9: Results from ATS Arabic-French. *65*
Figure 4.10 : Results from ATS Train Arabic-Test French. *65*
Figure 4.11: Results from ATS Train Arabic-Test English. *66*
Figure 4.12: Results from ATS Train French-Test English. *66*
Figure 4.13: Results from ATS Train French- Test Arabic. *67*
Figure 4.14: Results from ATS Train English-Test French**.** *67*
Figure 4.15: Results from ATS Train English-Test Arabic. *68*
Figure 4.16: Results of the Bleu score (English-English). *69*
Figure 4.17: Results of the Bleu score (English-French). *70*
Figure 4.18: Results of the Bleu score (English-Arabic). *71*
Figure 4.19: Results of the Bleu score (French-English). *71*
Figure 4.20: Results of the Bleu score (Arabic-Arabic). *71*
Figure 4.21: Results of the bleu score (Train Arabic-Test English). *72*
Figure 4.22: Results of the bleu score (Train English-Test Arabic). *73*
Figure 4.23: Results of the bleu score (train English-Test French). *73*
Figure 4. 24: Results of the bleu score (Arabic-Test French). *73*
Figure 4.25: Sentence BLEU score for validation dataset (Arabic-French). *74*
Figure 4.26: Sentence BLEU score for validation dataset (French-French). *74*
Figure 4.27: Results of the bleu score (train Arabic-Test Arabic). *75*
Figure 4.28: Sentence BLEU score for validation dataset (Train French-Test English). *75*
Figure 4.29: Sentence BLEU score for validation dataset (Arabic-English). *75*
Figure 4. 30: Sentence BLEU score for validation dataset (French-Arabic). *76*

LIST OF ACRONYMS

AL: Attention Layer..... 38
ATS: Abstractive Text Summarization..... 7
BLEU: BiLingual Evaluation Understudy 23
BOW: Bag Of words 32
FNN: Forword Neural Network..... 32
GRU: Gated Recurrent Unit 8
LSTM: Long Short Term Memory..... 8
MLP: Multi-Layer Perceptron..... 9
NLP: Natural Language Processing..... 8
RNN: Recurrent Neural Network 8
ROUGE: Recall-Oriented Understudy for Gisting Evaluation..... 24
Seq2Seq: Sequence to sequence 8
TS: Text Summarization..... 8

Chapter 1

Introduction

1.1 Introduction

Living in the information age has changed our environment. We are surrounded by data, not in a metaphoric sense; it is a physical reality. The Internet, television and radio signals are in circulation in the air and we receive them on our mobile devices everywhere. Data servers in the count of millions have connected all the planets like a spider web. Only Google has more than 1 million servers around the world that are online 24/7, storing new generated data. Faster telecommunication, higher bandwidths, and more advanced hardware, along with Web 3.0, have led to an exponential growth in digital data. A considerable portion of this stream of new data is textual.

It is no wonder that the web can be named the number 1 biggest resource for textual content, as it covers almost most of the available text corpus, like books, journals, and newspapers, e.g. in online libraries. It is also connected to millions of organizations, companies, universities, and research centers, and individual users that are creating content every day on websites, blogs, social networks, etc. While the world is covered with computers which communicate in their own language, we are still humans with biological brains. What can we do? How can we deal with living in this new world?

The idea is to decrease the volume. *Automatic Text Summarization* is an attempt to reduce the size of a corpus while keeping its valuable content. Mostly, the available methods of text summarization are either extractive or abstractive. In the extractive way, the system chooses a few sentences from a source corpus, as opposed to the abstractive way that breaks the corpus contents and decomposes them again in a summary. Indeed, abstractive summarization is very close to how the brain receives, processes, and regenerates information. As a result, it is very difficult to design and replicate an abstractive algorithm for automatic summarization. That is why, at the birth of the field in the 1950s, most of the methods fit into the extractive method.

The available algorithms, tools, and resources were not sufficient to create an abstractive summarization system.

To approach a problem, one can try to solve it directly, by designing algorithms, using available methods, techniques, and tools. This is like the disciplines that are used in Engineering. One might try to solve the problem indirectly. Indeed, he can use the available solutions and mimic them. It is very close to Reverse Engineering. It is also the procedure of discovering the systems and mechanisms that are already available in nature and benefiting them for our personal (as referred to as human) purposes.

In the Text Summarization terminology, the original summaries that are created by a human and are used as the metric for reassuming the accuracy of the summarizer system are called the Golden Set. Let us look at the problem from another angle. If the main criterion for creating a good summarizer system is that it be as good as a human, why do we not mimic a human approach in the first place? By this manifest, a Neural Text Summarizer has the potential to be the ultimate accurate text summarizer if we make it learn from a human style as accurately as possible.

1.2 Problem Background

Previous research has primarily focused on extractive text summarizing, in which significant words or passages are extracted from texts to provide an important summary, which is an accurate description of the content. Extractive text summarization has the flaw of extracting just the essential sentences from the text without doing an independent analytical summary. Now, more and more people are looking for the possibility of freer translations or summaries to solve the problem of just getting sentences from the source. They, like humans, utilize the original material as a summary to create a summary in their own terms.

In this thesis, a qualitative study was conducted to investigate the problem of Abstractive Text Summarization (ATS) using Attentional Encoder Decoder Recurrent Neural Networks as will be discussed in Chapter 3, The study suggested that we do not simply extract the existing sentences or words from the original text, instead we generate a highly compressed paraphrasing of the core information of the text, even leveraging vocabulary absent from the source if needed.

Abstract text summarization is defined as the generation of a short summary, which covers the core information of a passage or an article, consisting of a few keywords and sentences. So, attention mechanisms can be effective for selecting the most important words for a short summary while filtering out the negative impact of redundant sections in abstractive sentence summarization.

"One central component in state-of-the-art sequence to sequence models is the use of attention for building connections between the source sequence and target words, so that a more informed decision can be made for generating a target word by considering the most relevant parts of the source sequence" [1][2]

"The state-of-the-art in abstractive TS deep learning systems employ seq2seq models of encoder-decoder architectures along with attention mechanisms, primarily based on recurrent neural networks (RNNs) and especially on long short-term memory networks (LSTMs) and gated recurrent units (GRUs) " [3] [4] [5] [6] [7] [8]

1.3 Research Motivation

Text summarization in NLP has largely been limited to extractive methods that crop out and stick together portions of the original text in order to capture the bulk of the meaning concisely. What has been less attempted and is poorly understood are abstractive methods that might produce words not in the original text, giving more natural summaries and dynamic paraphrases text ,This research is motivated by the fact that it is challenging to create a short, fluent, and understandable abstractive summary of a text document. It is closer to what humans usually do.

Thus, this thesis proposes the use of *Amazon fine food reviews* [1] dataset, which is available on Kaggle. We have used reviews text descriptions as our input data, and generated a simple summary of those review descriptions as our output. To assist in producing some extensive summaries, we have used a bi-directional RNN with LSTM's in the encoding layer and an attention model in the decoding layer. And we applied the sequence to sequence model to generate a short summary of food descriptions.

1.4 Thesis Statement

A text summary is typically very short and does not depend very much on the length of the source (document) in summarization. Additionally, a key challenge in summarization is to optimally compress the original document in a lossy manner such that the key concepts in the original document are preserved. One major issue that remains unexplored is:

"After evolution in deep learning, can we get an abstractive summary of all the relevant information in the text? Can we really get an abstractive summary that has the power to sum up like the human brain? "

In this regard, this thesis addresses the problem of *Neural abstractive text summarization,* dedicated to tackling the problem of representing the diversity of information about the world in a meaningful, intuitive, and robust way. As can be expressed mathematically by the following:

[1] https://www.kaggle.com/snap/amazon-fine-food-reviews

Generic Problem: Given a piece of review, we would like to generate a shorter version of the review while preserving the sentiment and points.

More formally, Let the input sequence of M words be x_1, ..., x_M coming from a vocabulary of size V. We want to generate a shortened sequence y_1, ..., y_N such that N < M and *y* retains the essence of *x*. We assume al, $y_1,...,y_N$ coming from the same vocabulary.

The proposed solution is the attention mechanism used for abstractive text summarization and develops three modified techniques using attentional encoder-decoder recurrent neural networks that improve the abstractive summary (Rush et al.,2015). The model has to first truly understand the document and then try to express that understanding in short, possibly using new words and phrases. It has complex capabilities like generalization, paraphrasing and incorporating real-world knowledge. The corpus is compressed in a lossy manner while maintaining the meaning and salient features of the original data.

1.5 Decision Problems

To precisely formulate the research problem and organize the rest of the thesis, the following decision problems are introduced. In this context, we state each problem and give a research question to be answered throughout the thesis.

Problem 1: The core idea here is if output depends on a sequence S of inputs then we will choose a type of neural network which gives importance to sequence information, which somehow retains and leverages the sequence information.

Techniques like multi-layer perceptron (MLP) work well when your input data is a vector, and recurrent neural networks (RNN) work very well if your input data is an image.

Research Question:

"What if my input x is a sequence? What if x is a sequence of words? In most languages, the sequence of words matters a lot. Which kind will give us better results? "

Problem 2: To summarize, we create a semantic representation of the document in our brains. We then pick words from our general vocabulary (the words we commonly use) that fit the semantics, to create a short summary that represents all the points of the actual document.

Research Question:

"*How do we overcome this problem of making it difficult for the encoder to memorize long sequences into a fixed-length vector?****"***

. **Problem 3:** Finding informative words for a more comprehensive representation of the input sequence, which leads to a better summary, is difficult/tricky/hard. So instead of looking at all the words in the source sequence, we can increase the importance of specific parts of the source sequence that result in the target sequence.

Research Question:

"How much attention do we need to pay to every word in the input sequence for generating a word at timestep t?"

Problem 4: We used the model for machine translation. For summary generation, we need to compress the original document to create the summary, so we need the loss to be a bit lossy, unlike in machine translation.

Research Question:

"How can we minimize the loss and save time in memory as well as computation?"

1.6 Thesis Objectives

Based on our decision problems, this research aims to accomplish the following objectives:

I. To create an abstractive summarizer based on the concept of deep learning. The idea of a deep learning technique is to educate machines with some sample data so that they can produce recapitulation of input text.
    a) *To build the model while increasing the size of the training dataset. The ability of a deep learning model to generalize improves as the size of the training dataset rises.*
    b) *To generate a shorter version of the review while preserving the sentiment and points.*
    c) *We apply off-the-shelf attentional encoder-decoder RNN that was originally developed for machine translation to perform summarization.*

II. To develop RNN encoder-decoder that consists of two RNNs:

a) *The encoder extracts a fixed length representation of the input.*
b) *The decoder generates a translation from this representation.*
c) *We allow the decoder to have more direct access to input by implementing an attention mechanism.*

III. We tested the model on new source sequences for which the target sequence is unknown. So, we need to set up the inference architecture to decode a test sequence:

a) *The decoder is used to predict the probability of a word being selected.*
b) *We pass the word with the maximum probability to the decoder.*
c) *We generate the word that has the highest probability.*

1.7 Thesis Scope

The awesome concept of Text Summarization using Deep Learning really helped us out. It solves the issue of understanding large text documents and assimilating crucial information out of them is often a laborious and time-consuming task. – Now our model can understand the context of the entire text. It's a dream come true for all of us who need to come up with a quick summary of a document!

1.8 Thesis Outline

The thesis tackles the problem of abstractive text summarization using Attentional Encoder-Decoder Recurrent Neural Networks. To build the model has to truly understand the document and then try to express that understanding in words, possibly using new words and phrases. It has complex capabilities like generalization and incorporating real-world knowledge. The research achieved in this thesis is organized into the following chapters.

Chapter 2: Overview of Automatic text summarization, methods, techniques

This chapter discusses the *frameworks* for automatic text summarization in general. We've gone over what ATS is, how it works, and what its characteristics are. We've also gone over the different classifications for a summary based on their input, output, content, purpose, and language. We also discussed update summarization and the concept of significant phrases, as

well as a variety of techniques. Finally, some evaluation methods were introduced, which could be used to examine and compare the results of different approaches.

Chapter 3: Methodology

The chapter describes the methodology undertaken to tackle the four decision problems defined in the first chapter. This chapter introduces the methodology which will be followed to achieve the desirable goal. A methodology is generally a guideline for solving a research problem. It comprises of a generic research framework and the steps required to conduct the research in a planned manner. It also includes discussion of the major research components, such as the phases, and the techniques involved.

Chapter 4: Test and Evaluation

The results of our approach performance evaluation are provided in this chapter with discussion and comparison with abstractive summarization in 3 languages (Arabic, English, French).

Chapter 2

Automatic text summarization

2.1 Introduction

Due to the great amount of information we are provided with and the huge number of articles which are published every day, investigating and monitoring of all the topics are not possible in a short time, therefore the need of producing summaries have become more and more widespread.

A summary can be defined as a text that is produced from one or more text(s), that contains a significant portion of information in the original text(s).

Text summarization is the process of extraction of the most important information from a source to produce an abridged version for a particular user and task. When this is done by means of computer i.e automatically, we call this Automatic Text Summarization.

2.2 Definition of Text Summarization

The literature provides various definitions of text summarization:

DEFINITION 1.1: According to Radev et al, « a summary is a text produced from one or more texts, which contains important and significant information in the original text(s), and this does not represent more than half of the original text(s) and generally less than that». [9]

DEFINITION 1.2: Radev et al add that : Text Summarization (TS) is the process of identifying salient concepts in text narrative, conceptualizing the relationships that exist among them and generating concise representations of the input text that preserve the gist of its content .[10]

According to Horacio Saggion and Guy Lapalme , in terms of function, a summary is: **DEFINITION 1.3:** a condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source [10].

2.3 The need for automatic summarization

With the huge amount of information generated every day in form of web pages, articles, etc. It is difficult to find the information desired by the manual's ways.

The purpose of automatic text summarization is to extract the main points from the original text without having to read the entire document while selecting the most salient sentences, minimize information redundancy and make the summary readable and understandable, which allows:

- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarizers.
- Personalized summaries are useful in question-answering systems as they provide personalized information. [10].

2.4 Related Works

Since we are implementing a neural network-based solution, we will be looking into other solutions here. There have been several solutions using neural networks for summarization.

In 2004, Kaikhah [24] proposed a system to extractive summarize text. This system generates seven features for each sentence in the document. The first four features represent the placement of the sentences in the document. These features encode whether the sentence or paragraph is the first one, and its relative position. The rest of the features express sentence length and content with respect to the most used words in the document and the title. After these features are generated, a neural network is trained to classify the sentences that should appear in the summary. Then the system is modified to generalize and filter the sentences that belong to the summary. This approach mixes feature engineering with machine learning.

Rush et al [25] in 2015 proposed a neural network-based system for abstractive sentence summarization with attention. In their abstract, they built a simple system that can scale with data and train end-to-end that showed significant performance gains. This system has a neural network to score the next word given the previous words as the generation step and has another encoder for information encoding. These two models are trained jointly and then beam searching is used to generate the final summary.

In 2016, building on research from Rush et al[25], Chopra et al [26] built a system for abstractive sentence summarization with attention using recurrent neural networks. This system improves on Rush et al [25] research. The main contribution of this model to Rush et al.'s research is its use of a convolutional attention-based recurrent encoder.

Also, in 2016, Nallapati et al [27] proposed a system for document level abstractive summarization using a seq2seq network. This system outperformed previous state-of-the-art systems on datasets it was applied to. Later in 2017, Nallapati et al [28] released another system that improved on 2016 research. This newer system could be trained both extractively and abstractively. While abstractive systems outperformed the current state-of-the-art, extractive sstems performed better.

2.5 Characteristics of a summary

A summary must have some characteristics: conciseness, coverage, fidelity, cohesion and consistency.

We expose these characteristics as they were defined in [12]:

- **Conciseness:** it is related to the reduction rate which is the ratio between the length of the source text and length of the summary. In general, the reduction rate is proportional to the restrictive nature of criteria used to generate the summary. For example, an indicative summary produces fairly short summary that includes just the main idea of the text. However, an informative summary contains more details about the subject and will in most cases lead to longer summaries.
- **Loyalty:** it is also an important criterion for characterizing a summary. It represents the relationship of objective similarity between the summary and the source text. It is a sort of measure of the overall quality of the summary. The notion of fidelity includes coverage as a component. As a general rule, a summary with correct coverage will be fairly faithful to the source text.
- **Coverage:** is a kind of ratio between the number of themes or elements present in the source text and those present in the summary. The nature of the elements depends on the type of summary considered: for an indicative summary, only the themes addressed will be retained. In the case of an informative summary, coverage is more difficult to determine.

- **Cohesion and consistency:** The last two criteria defining a summary are closely linked to the notion of text itself. Cohesion can be seen as the result of the application of mechanisms aimed at maintaining a referential and argumentative (use of connectors) unity. Coherence, on the other hand, derives more from the correct application of rhetorical.

2.6 Steps of automatic text summarization

There are three main steps to summarize a document: identification, interpretation and generation of summary [13].

- **Identification:** In this step, the most important information in the text are identified. There are different techniques for identifying subjects; methods based on the position of sentences are the most useful methods for identifying subjects.
- **Interpretation:** the important information that are identified in the first step are combined more cohesively. During this step, some modifications of the original sentences may prove necessary.
- **Generation:** the result of the second step is a summary which may not be consistent to the reader. Consequently, the aim of this step is to reformulate the extracted summary into a new coherent and understandable text.

2.7 Concept of salient sentences and scoring sentence

A principal concern in extractive document summarization is the selection of the most important content that can represent a document for inclusion in the summary output. Those sentences are known as salient sentences.

The focus of these research areas is addressed by the following question: how can a system determine which sentences are representative of the content of a given text? In general, three approaches are followed:

- Word scoring – assigning scores to the most important words.
- Sentence scoring – verifying sentences features such as its position in the document, similarity to the title, etc.
- Graph scoring – analyzing the relationship between sentences.

Based on those sentences, we can evaluate the pertinence of a summary. To do that, different ranking algorithms have been used to decide which sentences are more important and tend to be selected as summary sentences by assigning a pertinence score for each sentence. The score for each sentence is simply the linear combination of the weights given for each feature. The highest scoring sentences are selected as candidates to be part of the summary in the last stage.

The following section presents the main methods in each of the aforementioned approaches [14]:

1- Word scoring: The initial methods in sentence scoring were based on words. Each word receives a score and the weight of each sentence is the sum of all scores of its constituent words. The approaches in the literature are: word frequency, TF/IDF, number of upper-case word and proposer nouns, word co-occurrence and lexical similarity.

2- Sentence scoring: This approach analyzes the features of the sentence itself and was used for the first time in 1968 (Edmundson, 1969) by analyzing the presence of cue words in sentences. The main approaches that follow this idea are: cue-phrases, sentence inclusion of numerical data, sentence length, sentence position, sentence centrality, sentence resemblance to the title.

3- Graph scoring: In graph-based methods, the score is generated by the relationship among the sentences. When a sentence refers to another it generates a link with an associated weight between them. The weights are used to generate the score of sentences using: text rank, Bushy path of the node and Aggregate similarity.

2.8 Various types of summaries

Automatic summaries and their methods can be categorized according to different criteria. We will quote the most known and used ones [15]:

2.8.1 Summary based on input

In terms of input, a summary may be based on:

A) Single document: early attempts in summarization were mainly based on a single document summary, in which systems produced a summary from a single document.

B) Multiple documents: the system generates a summary for a set of documents that share a similar subject. This type is more difficult than the single document. Indeed, the system should remove any redundancies between documents and also reconcile the content into a coherent summary.

2.8.2 Summary based on details

A) Indicative summary: an indicative summary system gives a global perspective of the subjects covered by the text; it presents only the most important idea of the text. This type of summary helps the user to decide on whether or not to read the document, indicating the themes addressed and developed in the source document, without considering the details. Summary's length depends on the compression rate.

B) Informative summary: the informative summary system is considered to be an abridged version and covers all aspects of the main text preserving the general organization of the source document. Its main objective is to inform the reader about the main quantitative and qualitative information of the text. The length of the informative summaries is approximately 20-30% of the original text.

2.8.3 Summary based on content

A) Generic summary: aims to summarize the source text without noticing its domain and context.

B) Query-based summary: the summary is generated by selecting a sentence which corresponds to the user's query. Sentences that are relevant to the query receive a higher chance of being retrieved for the final summary.

Query-based summary systems, however, do not provide an overall view of document's concepts because they focus on the user's query.

C) Domain specific summary: it provides a summary according to the specific domain, for example: summarizing news articles, web pages and biomedical documents. This type of summary requires domain-specific knowledge to select sentences for summaries.

2.8.4 Summary based on output

A) Extractive summary: an extractive summary is generated by selecting the relevant and important sentences as they appear in the source document and concatenating them without any modification. It is a simple and robust way to produce a summary.

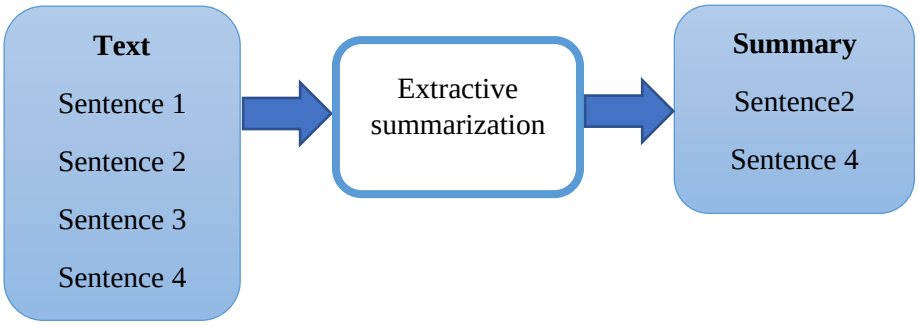


Figure 2.1 : Extractive text summarization.

B) Abstractive summary: in this type of summary, natural language generation techniques are used. We try to understand the original document by identifying the key concepts and then produce new sentences that are grammatically correct, concise, consistent and which should give a result close to a human summary.

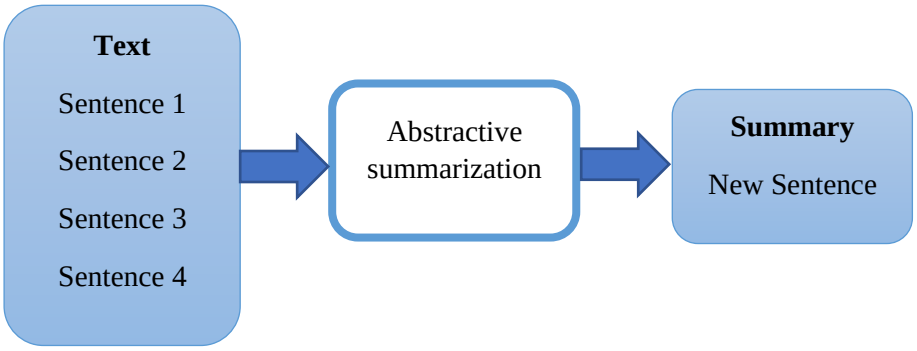


Figure 2.2: Abstractive text summarization.

C) Hybrid summary: This type of summary merges between both the extractive and abstractive approaches to generate the output text.

2.8.5 Summary based on language

A) Mono-lingual summary: the source document and the summary are written in the same language.

B) Multi-lingual summary: the system can process multiple languages, and produce summaries in the same language as the source document.

C) Cross-lingual summary: the language of source document and the summary must be different.

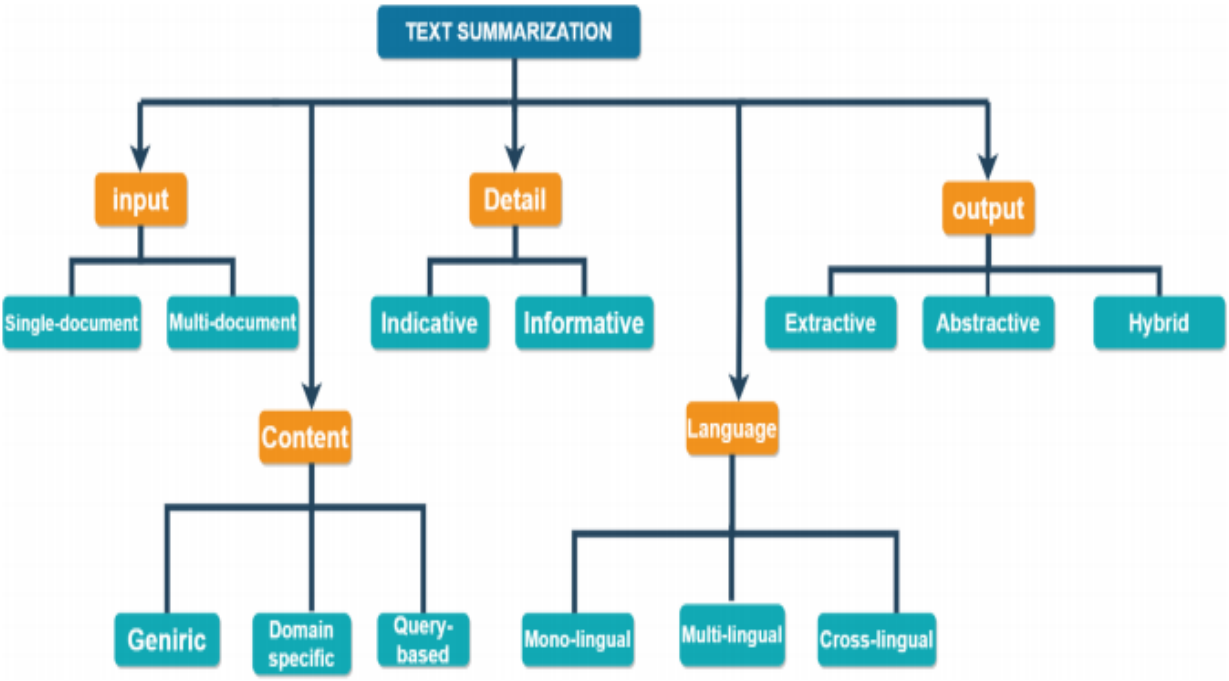


Figure 2.3: Text summarization classification.

2.9 Update summarization

With the ever-increasing popularity of news search engines, displaying the information in a more practical and pleasant way is becoming a challenging and important issue. One possible solution is to summarize multiple news so as to propose only one short text. This is, intuitively, a reasonable solution though producing summaries from large collection of documents is a very complicated task. However, as the number of documents increases, facts that are considered as important and have to appear in the summary also become more numerous. In this case, a choice must then be made to drop important facts in order to satisfy size constraints. One way to tackle this problem is to remove facts that the user is already aware of. This variant of text summarization is called update summarization. More formally, update summarization is the task of producing summaries while minimizing redundancy with previously read documents. Indeed, segments have to be selected according to their salience but also to their ability to capture novelty.

The most intuitive way to go about update summarization would be to identify temporal references within documents (dates, elapsed times, temporal expressions, etc.) and to construct a time-line of the events.

It is a complex task as temporal references depend on surrounding elements in the discourse but also require an understanding of the ontological and logical foundations of temporal reference construction. Assuming the time-line is constructed, update summaries could be produced by assembling sentences containing the most recent events [16].

2.10 Text summarization methods

The five main approaches for summarization are the statistical, the graph-based, machine learning and swarm intelligence approaches, Semantic based approaches.

- **Statistical based Approaches:** were the first introduced in ATS. These techniques are independent of any language, then it can summarize text in any language. They concentrate on statistics got from non-linguistic features of the document, like term frequency, position and length, cue phrases, title words, resemblance of sentence to the title and sentence position.

  A statistical approach does not require any training dataset and does not need much language dependent tools; just some basic NLP tasks such as sentence segmentation, tokenization, stop word elimination, and stemming. Mostly, it is easy to be implemented and does not need a lot of processing power. So, for each sentence in the document, a score is computed and highly scored sentences are chosen for generating the summary, mostly these features are combined together in hope to increase sentence relevance [17].

- **Machine learning based approaches:** learn from the data. They can be supervised, unsupervised or semi-supervised .In supervised approach, there is a collection of documents and their respective human-generated summaries such that useful features of sentences can be learnt from them. Supervised or trainable summarizers classify each sentence of the test document either into *«summary»* or *«non-summary»* class with the help of a training set of documents.

  Large amount of labeled or annotated data is needed for learning purpose. Support

Vector machine (SVM), Naïve Bayes classification, Mathematical Regression, Decision trees, Neural networks (Multilayer Perceptron), are some of the supervised learning algorithms. On the other hand, unsupervised systems do not require any training data. They generate the summary by accessing only the target documents. They try to discover hidden structure in the unlabeled data. Thus, they are suitable for any newly observed data without any advanced modifications. Such systems apply heuristic rules to extract highly relevant sentences and generate a summary.

Semi-supervised learning techniques require labeled and unlabeled data both to generate an appropriate function or classifier [18].

- **Semantic based approaches**: semantic approaches mainly focus on the semantic analysis and correlations among sentences. They identify the relationship between words and sentences by use of thesaurus, ontologies, part of speech tagging, grammatical analysis and selection of meaningful sentences to generate summary. Various techniques like lexical chain, cluster and graph-based methods have been developed in this approach. Semantic-based methods are useful since they consider the meaning of each sentence and word that make a coherent and meaningful summary but using these techniques are time-consuming and require more efforts than the other techniques [13].

- **Graph based method:** In the field of natural language processing, graphs are used to display the structure of the text and the connection between sentences. Sentences are represented as a node, and the relation between sentences are depicted by edges. The graph-based text summarization method is a technique to extract a significant, appropriate, and informative text in a compressed version. In order to use this technique, a pre-processing phase should be done on the input text to remove stop words, tokenize the sentences, and so on. Then sentences are ranked to identify the important sentences. Afterward, the relation between sentences is computed to recognize the relevant sentences. At the end, sentences are extracted for the summary based on the ranked and relevant sentences [19].
- **Swarm intelligence approaches:** in a computational context, a swarm is a group of simple agents that have a collective behavior to perform a complex task by acting as a community. Swarm Intelligence algorithms have recently sprung up as a family of nature-inspired, population methods that are capable of producing low cost, fast, and

robust solutions to several complex problems. They can, therefore, be defined as a relatively novel branch of Artificial Intelligence (AI) that is employed to model the collective behaving of societal swarms in nature, for example the social behavior of ants, termites, bees, and other social beings have motivated researchers to discover their lifestyle, they noticed that they are moderately innocent with restricted capacities on their own, they are interacting together with certain behavioral patterns to cooperatively accomplish tasks necessary for their survival. The social connections among swarm individuals can be either direct or indirect. Some swarm-based algorithms such as particle swarm optimization, cuckoo optimization algorithm, and bacterial foraging optimization have been introduced in text summarization [20].

Approach	**Method**	**Advantages**	**Disadvantages**
Statistical	-Cue phrase -Title words -Sentence location -Word frequency	-Simple to perform	-Don't consider the meaning of word on the text. -Duplication in summary sentences.
machine learning	-Naive bases method -Artificial neural network - Fuzzy logic	-Comprehensive summary is produced.	-Depend on the training dataset. -Complexity in computation
semantic based	- Lexical chain - Clustering - Graph based method	-Reduce redundancy in multi-documents summarization.	- Limit each sentence to be put only in one cluster
Swarm intelligene	- Particle swarm optimization. - Cuckoo optimization algorithm. - Bacterial foraging optimization method.	-Better performance compared with other. method	- Complexity in computation.

Table 2.1: Comparison between different Automatic Text Summarization approaches that exist according to their advantages and disadvantages.

2.11 Evaluation techniques

Evaluation methods can be classified into two types intrinsic and extrinsic method:

2.11.1 Intrinsic evaluation

Its purpose is to evaluate the performance of automated summary content by comparing it with an ideal summary, it has assessed mainly the coherence and informativeness of summaries. There are two metrics for the intrinsic metric. First measure involves quality evaluation, which tries to verify that the summary is syntactically correct, has no grammatical errors, logically articulated, has structural consistency and does not contain redundant information. Second measure is content evaluation which are divided into two groups: co-selection measure and content-based metric [21].

The co-selection measure can count as a match only exactly the same sentences. This ignores the fact that two sentences can contain the same information even if they are written differently. Furthermore, summaries written by two different annotators do not in general share identical sentences. It consists of a precision, recall, and F-measure.

- **Precision:** it reflects how relevant the selected data is. It is computed by the intersection of summarized extracted, and ideal sentences, divided by all extracted sentences.

- **Recall:** is computed by the intersection between the relevant and retrieved sentences, divided by all the relevant sentences.

- **F-measure:** is an average of precision and recall criteria, and determines the score of the final set of the selected sentences in a produced summary.

On the other hand, when we defined abstractive and extractive summarization, we used a score function. However, we did not define what it was. In fact, it is quite hard to define. One way to define it is, given a model θ, by using the log probability of a summary given a document and a model:

$$s(x,y) = log\big(p(y|x;\theta)\big) \approx \sum_i log\left(p\big(y_{i+1}\big|x, y_{[0:i]}, \theta\big)\right)$$

where $y_{[0:i]}$ are all previous outputs before output i. Here, we make the assumption that a word in the summary is only determined by all previous words and the input text. What was done in traditional summarization was to separate the probability into two parts by using Bayes' rule:

$$argmax\left(log\big(p(y|x)\big)\right) = argmax\left(log\big(p(x|y)\big)\right)$$

where $p(y)$ is the language model and $p(x|y)$ is the summarization model? In this work, we will approximate $p(y|x)$ directly with our neural network. We have just defined a score function given a model. However, the way summaries are evaluated in the general case is not clear. For summarization, we can use the measurements which are used in machine translation: **BLEU** and **ROUGE** scores.

BLEU (BiLingual Evaluation Understudy) [22] is a precision-oriented score. It is defined as follows:

$$\text{BLEU}= \frac{number\ of\ words\ in\ the\ summary\ which\ are\ in\ the\ gold\ standard}{Total\ number\ of\ words\ in\ the\ summary}$$

The recall side of **BLEU** is called **ROUGE** (for Recall-Oriented Understudy for Gisting Evaluation). There are different versions of ROUGE. The direct BLEU equivalent is ROUGE-1 and counts unigram overlaps:

$$\text{ROUGE-1}=\frac{\sum_{Reference\ Summary}\sum_{Unigram} count_{match}(unigram)}{\sum_{Reference\ Summary}\sum_{Unigram}\ count\ (unigram)}$$

This definition can be generalized to N-grams:

$$\text{ROUGE-N}=\frac{\sum_{Reference\ Summary}\sum_{N\ gram} count_{match}(N\ gram)}{\sum_{Reference\ Summary}\sum_{N\ gram}\ count\ (N\ gram)}$$

There exist other extensions of ROUGE. For instance, ROUGE-L takes into account the longest common sequence and Rouge-S and Rouge-SU consider skip sequences.
Evaluating a summary is still a topic of research and there is no perfect consensus to do it yet.

• **ROUGE-L:** This measure employs the concept of longest common subsequence (LCS) between the two sequences of text. The intuition is that the longer the LCS between two summary sentences, the more similar they are. Although this metric is more flexible than the previous one, it has a drawback that all n-grams must be consecutive.

• **ROUGE-SU:**This metric called skip bi-gram and uni-gram ROUGE and considers bi-grams as well as uni-grams. This metric allows insertion of words between the first and the last words of the bi-grams, so they do not need to be consecutive sequences of words.

2.11.2 Extrinsic evaluation

The extrinsic evaluation metric is known as task-oriented (task-based) metrics. These metrics do not analyze sentences in the summary. They try to measure the prospect of using summaries for a certain task.

Various approaches to task-based summarization evaluation can be found in the literature. Question-answering, text classification and information retrieval are examples of the extrinsic method. Their objective is to evaluate a summary performance based on a special task [21].

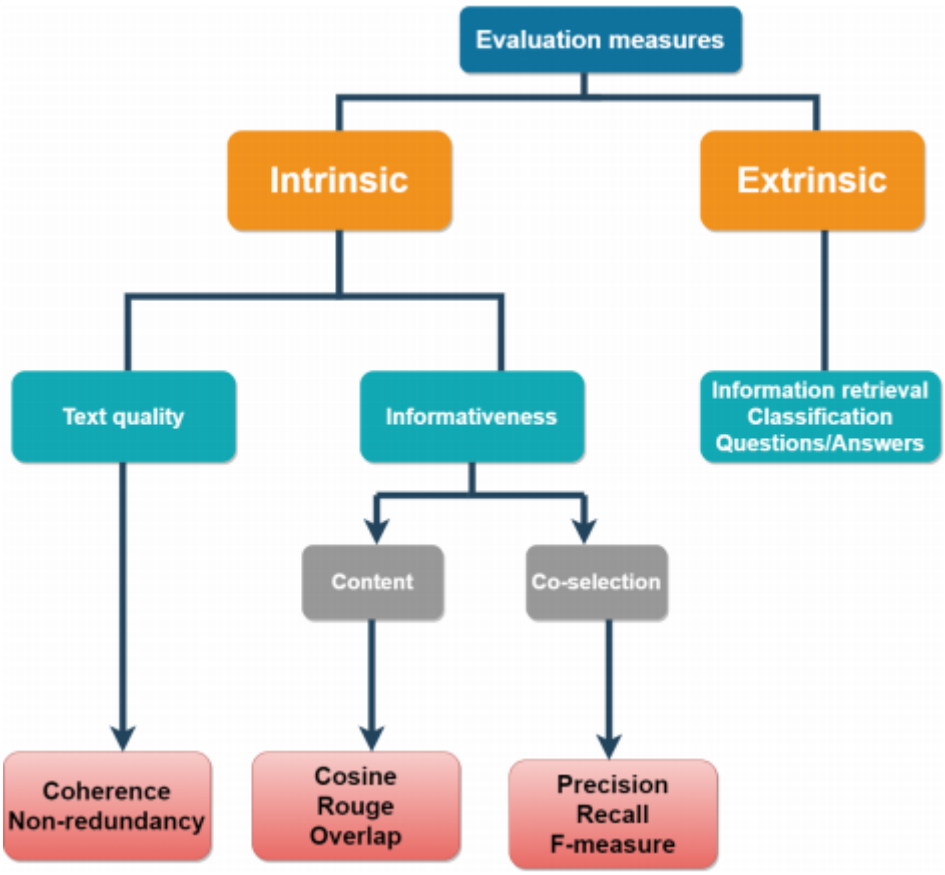


Figure 2.4: Evaluation techniques of Automatic Text Summarization.

In this part we focused on extractive summarization, even though summaries created by humans are usually not extractive, most of the summarization research today has focused on extractive summarization. Purely extractive summaries often times give better results compared to automatic abstractive summaries.

2.12 Summarization Main Problems

There are three main problems that need to be tackled to write a good summary

1. Content need to be *selected.* A text contains a lot of information. Some are redundant, some are less important than others and some are of no interest for the reader. Therefore, a method to select relevant contents from the ones which are not is required. Basically, this is a classification task.

2. Once the components of the summary are chosen, the information has to be *ordered.* In fact, if the information is not ordered correctly, it may change the meaning of the original text. So one needs to be careful with it. For single document summarization, one may assume that the order of sentences in the original text implies the order of the information but for multi-document summarization, the problem is not trivial.

3. Sentences need to be *generated*. This step is important to make the sentences easy to read for human. The final result has to look fluent. It is also important to correctly express the original content. Another aspect of this stage is to compress: the information needs to be expressed the most efficiently possible.

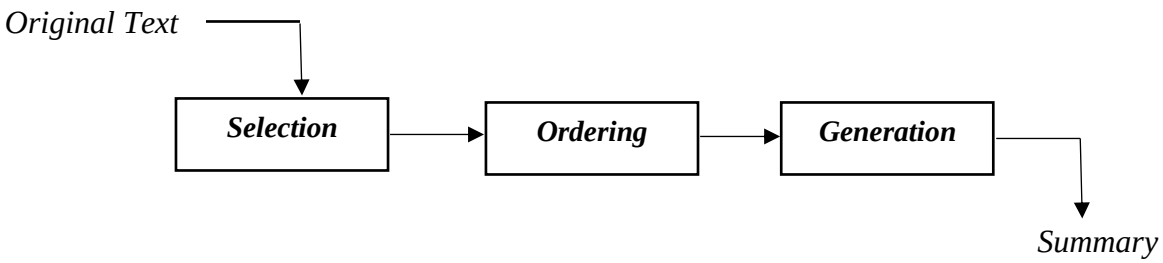


Figure 2.5: Main steps of summarization.

2.13 Conclusion

In this chapter, we have explained what does Automatic Text Summarization mean and related work and what are its characteristics, we have given an overview about different classifications for a summary based on their input, output, content, purpose and language. We have also talked about the update summarization and the concept of salient sentences, then a various technique like statistical, machine learning, semantic-based and swarm intelligence-based methods were described. Finally, some evaluation method was introduced, which could be used to examine and compare the results of different approaches.

In this work, we focus on improving abstractive methods, because of their ability to generate novel sentences. Knowing what the existing summarization systems are able to offer and what issues they are facing can help future design decisions made much more wisely.

Chapter 3

Methodogy

3.1 Introduction

Text summarization is one of the renowned issues in natural language processing and deep learning in recent years. Text summary usually consists of a brief note on a huge text document. Our major goal is to generate an abstractive summary of a text document that is brief, fluent, and comprehensible. We used the Amazon fine food reviews dataset to create the best possible summarizer.

3.2 Research Design

In this thesis, we have conducted scientific and applied research on abstractive text summarization. The research was developed using deep learning, we have used reviews text descriptions as our input data, and generated a simple summary of those review descriptions as our output. To assist in producing some extensive summaries, we applied the sequence to sequence model to generate a short summary of food descriptions, and we have used a bi-directional RNN with LSTM's in the encoding layer and an attention model in the decoding layer.

There are some challenges when working with abstractive text summarizers, such as text processing, vocabulary counting, missing word counting, word embedding, the efficiency of the model or reducing the value of loss and response machine fluent summary. In this thesis, the main goal was to increase the efficiency and reduce train loss of the sequence to sequence model to make a better abstractive text summarizer.

3.3 General Research Framework

The general workflow of this research was conducted under five phases:
Analysis, **D**esign, **D**evelopment, **I**mplementation and **E**valuation, as illustrated by Figure 3.1.

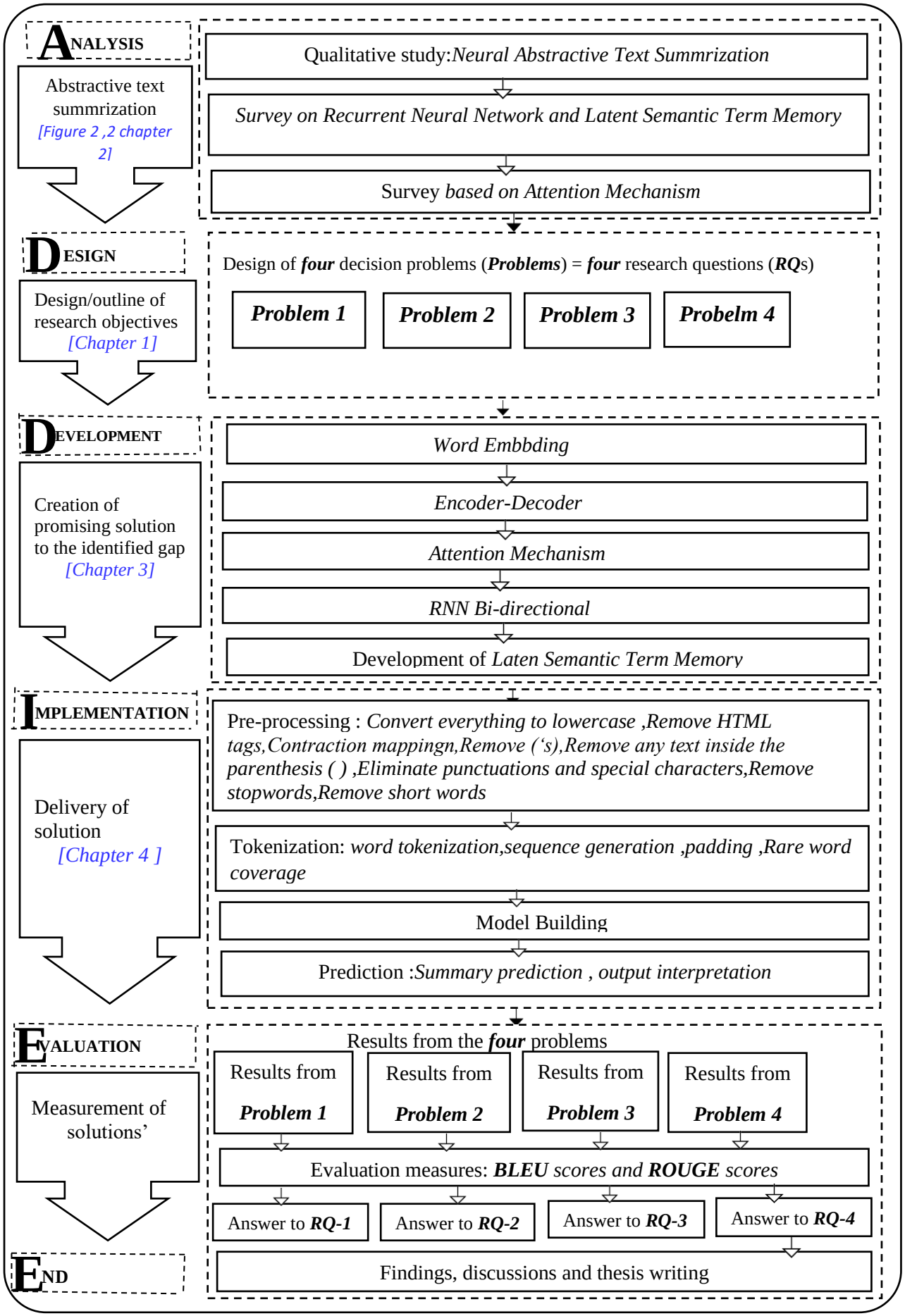


Figure 3.1: Overall research framework and activities implemented in the thesis.

Phase 1: Analysis

This phase involved a qualitative study conducted to investigate the Neural Abstractive summarization the detailed discussion is presented in Chapter 3 and 4. Also, the analysis phase includes an extensive and broad literature review on Automatic summarization as discussed in Chapter 2. Another survey was dedicated principally to semantic-based text similarity measures, as presented at the end of Chapter 2.

Phase 2: Design

This phase involved the design of four decision problems that we mentioned previously, and we will be investigating to find the answers.

Phase 3: Development

The development phase introduced Word embedding is one of the most popular representations of document vocabulary. It is capable of capturing the context of a word in a document, semantic and syntactic similarity, relationships with other words, etc. A word-embedding is used to represent words input to the encoder. This is a distributed representation where each word is mapped to a fixed-sized vector of continuous values. The benefit of this approach is that different words with similar meanings will have similar representations. The decoder is trained to start generating the output sequence depending on the information encoded by the encoder. These methods have been used widely.

Attention is an interface connecting the encoder and decoder that provides the decoder with information from every encoder's hidden state. In RNN, the new output is dependent on the previous output. Due to this property of RNN, we try to summarize our text as more human as possible. LSTM is capable of learning long-term dependencies, thus treating the problem of short-term dependencies of a simple RNN.

Phase 4: Implementation

The implementation phase demonstrated the steps for generating an abstractive summary, which we will explain in detail next in this chapter.

Phase 5: Evaluation

In the evaluation phase, we considered the use of BLEU measures precision: how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries.

ROUGE measures recall: how much the words (and/or n-grams) in the human reference summaries appeared in the machine-generated summaries.

Naturally, these results are complementing, as is often the case in precision vs recall. If you have many words from the system results appearing in the human references, you will have high Bleu, and if you have many words from the human references appearing in the system results, you will have high Rouge.

3.4 Dataset

This dataset consists of reviews of fine foods from Amazon. The data spans a period of more than 10 years, including all 500,000 reviews up to October 2012. These reviews include product and user information, ratings, plain-text reviews, and a summary. It also includes reviews from all other Amazon categories. We took a sample of 200,000 reviews.

- **Reviews dataset**

	Summary	Text
0	Good Quality Dog Food	I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labr...
1	Not as Advertised	Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as "Jumbo".
2	"Delight" says it all	This is a confection that has been around a few centuries. It is a light, pillowy citrus gelatin with nuts - in this case Filberts. And it is cut into tiny squares and then liberally coated with ...
3	Cough Medicine	If you are looking for the secret ingredient in Robitussin I believe I have found it. I got this in addition to the Root Beer Extract I ordered (which was good) and made some cherry soda. The fl...
4	Great taffy	Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery was very quick. If your a taffy lover, this is a deal.

Figure 3.2: Reviews dataset.

- **Reviews dataset (translated to Arabic)**

	S_ar	T_ar
0	طعام كلاب ذو نوعية جيدة	ووجدتها كلها ذات نوعية جيدة. المنتج يشبه الحساء أكثر من اللحوم المصنعة وبراحته أفضل. لابرادور الخاص بي صعب وهي تقدر هذا المنتج أفضل من Vitality لقد اشتريت العديد من منتجات طعام الكلاب المعلبة معظم...
1	ليس كما هو معلن	."وصل المنتج المسمى بالفول السوداني الجاميو المملح ... كان الفول السوداني صغير الحجم غير مملح. لست متأكدًا مما إذا كان هذا خطأ أو إذا كان البائع ينوي تمثيل المنتج على أنه "جاميو
2	البيجة" تقول كل شيء"	ويتم تقطيعها إلى مربعات صغيرة ثم يتم تغليفها بمسحوق السكر. وهي جرعة صغيرة من السماء. .Filberts هذه حلوى كانت موجودة منذ عدة قرون. إنه جيلاتين حمضيات خفيف ومريح مع المكسرات - في هذه الحالة ليس مطاط...
3	دواء السعال	.أعتقد أنني وجدته. حصلت على هذا بالإضافة إلى مستخلص جذر الجذر الذي طلبته (والذي كان جيدًا) وصنع بعض صودا الكرز. النكهة طيبة جدا ، Robitussin إذا كنت تبحث عن المكون السري في
4	حلوى كبيرة	.حلوى كبيرة بسعر رائع. كان هناك تشكيلة واسعة من الحلوى اللذيذة. كان التسليم سريعًا جدًا. إذا كنت من محبي الحلوى ، فهذه صفقة

Figure 3. 3: Reviews dataset (translated to Arabic).

- **Reviews dataset (translated to French)**

	S_fr	T_fr
0	Nourriture pour chiens de bonne qualité	J'ai acheté plusieurs aliments en conserve pour chiens Vitality et je les ai tous trouvés de bonne qualité. Le produit ressemble plus à un ragoût qu'à une viande transformée et il sent meilleur. M...
1	Pas comme annoncé	Le produit est arrivé étiqueté comme étant des arachides salées géantes. . . les arachides étaient en fait de petite taille non salées. Je ne sais pas s'il s'agissait d'une erreur ou si le vendeur...
2	"Délice" dit tout	C'est une confection qui existe depuis quelques siècles. C'est une gélatine d'agrumes légère et moelleuse avec des noix - dans ce cas, des avelines. Et il est coupé en petits carrés puis généreuse...
3	Médicaments contre la toux	Si vous cherchez l'ingrédient secret du Robitussin, je pense l'avoir trouvé. Je l'ai obtenu en plus de l'extrait de racinette que j'ai commandé (ce qui était bon) et j'ai fait du soda à la cerise....
4	Excellente tire	Excellente tire à un bon prix. Il y avait un large assortiment de délicieuses tirees. La livraison a été très rapide. Si vous êtes amateur de tire, c'est une bonne affaire.

Figure 3.4: Reviews dataset (translated to French).

- **Reviews dataset (translated to Arabic and French Summary)**

	T_ar	S_fr
0	ووجدتها كلها ذات نوعية جيدة. المنتج يشبه الحساء أكثر من اللحوم المصنعة ورائحته أفضل. لا يزال الدور الخاص بي صعب وهي تقدر Vitality لقد اشتريت العديد من منتجات طعام الكلاب المعلبة هذا المنتج أفضل من معظم...	Nourriture pour chiens de bonne qualité
1	".وصل المنتج المسمى بالفول السوداني الجاميو المملح ... كان الفول السوداني صغير الحجم غير مملح. لست متأكدًا مما إذا كان هذا خطأ أو إذا كان البائع يدوي تمثيل المنتج على أنه "جاميو	Pas comme annoncé
2	ويتم تقطيعها إلى مربعات صغيرة ثم يتم تغليفها بمسحوق السكر. وهي .Filberts هذه حلوى كانت موجودة منذ عدة قرون. إنه جيلاتين حامضيات خفيف ومريح مع المكسرات - في هذه الحالة جرعة صغيرة من السماء. ليس مطاط...	"Délice" dit tout
3	أعتقد أنني وجدته. حصلت على هذا بالإضافة إلى مستخلص جذر الجذر الذي طلبته (والذي كان جيدًا) وصنع بعض صودا الكرز. النكهة ، Robitussin إذا كنت تبحث عن المكون السري في عليه جدا.	Médicaments contre la toux
4	.حلوى كبيرة بسعر رائع. كان هناك تشكيلة واسعة من الحلوى اللذيذة. كان التسليم سريعًا جدًا. إذا كنت من محبي الحلوى ، فهذه صفقة	Excellente tire

Figure 3. 5: Reviews dataset (translated to Arabic, French).

- And we did the same thing to Reviews dataset (translated to Arabic, French) and Reviews dataset (translated to Arabic, English), Reviews dataset (translated to French, English).

3.5 Recurrent neural networks (RNN) for text summarization

RNNs [33] are used for sequential data which involves variable length inputs or outputs such as speech and language and they are adapted for processing data as a sequence [34].

In contrast to FNNs which commonly pass the input data directly from input to output nodes, RNNs have cyclic or recurrent connections among nodes of distinct levels. This makes possible to model the output of the network by taking into account the entire history of the received input data by maintaining in their hidden units an internal memory states or ‘state vector’. Recurrent connections connect past data with the one that is currently being processed, simulating a state memory. The forward pass is similar to the one in FNNs. The unique difference is that the activation of a node depends on both the current input and the previous status of the hidden layers. The text is fed into the network by means of vector representations that are recurrently processed. This means that RNNs view a sample text as an ordered sequence of word identifiers, differently from common FNNs working on hand-crafted inputs, e.g., Bag of Words (BOW). [35] [36] [37]

Figure shows a simple architecture of RNN where X_i are input tokens, $W_{xh}$are weights for the connections between input layer and hidden layers, W_{yh} are weights for connections between hidden layer and output layer, ai are activations of hidden layer neurons and $Y_i$are outputs.

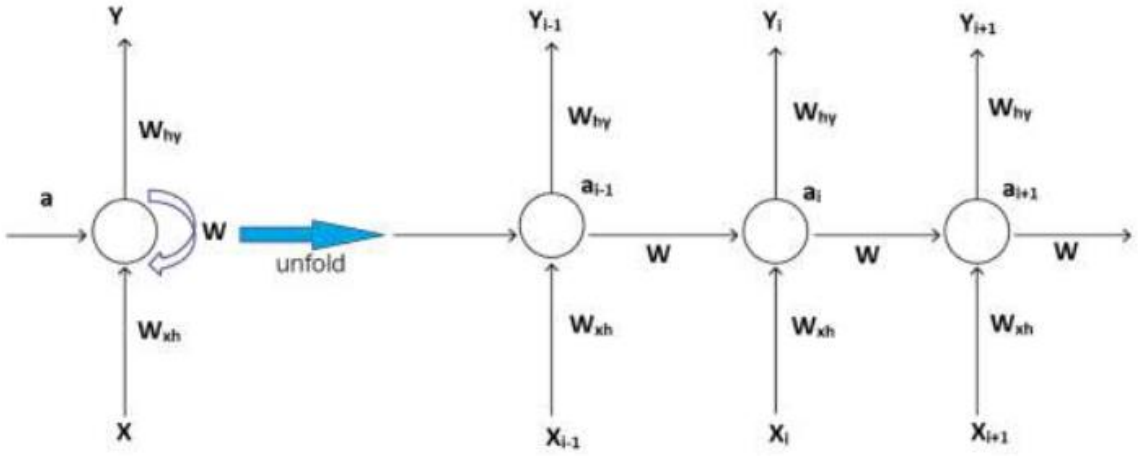


Figure 3.6:RNN Architecture.

In a wide range of applications, data can present patterns from the past to the future and vice versa. For instance, for classifying the sections of a given story, it could be useful to have access to both future and past sections. However, the future content of a text is ignored by common FNNs and RNNs, as they work sequentially.

Bidirectional RNNs (BiRNNs) let the network, at a given point in time, to take information from both earlier and later data in the sequence, going beyond the exposed limitations. The idea behind this kind of networks consists of presenting the training data forward and backward by two hidden RNNs which are then combined into a common output layer (Figure 10). This strategy makes possible to find patterns that can be learned from both past and future history of data [35].

Although the main purpose of RNNs is to learn long-term dependencies, theoretical and empirical evidence shows that it is difficult to learn and to store information for very long. In addition, the training of RNNs proved to be problematic because the backpropagated gradients either grow or shrink at each time step, so over many time steps they typically explode or vanish [38].

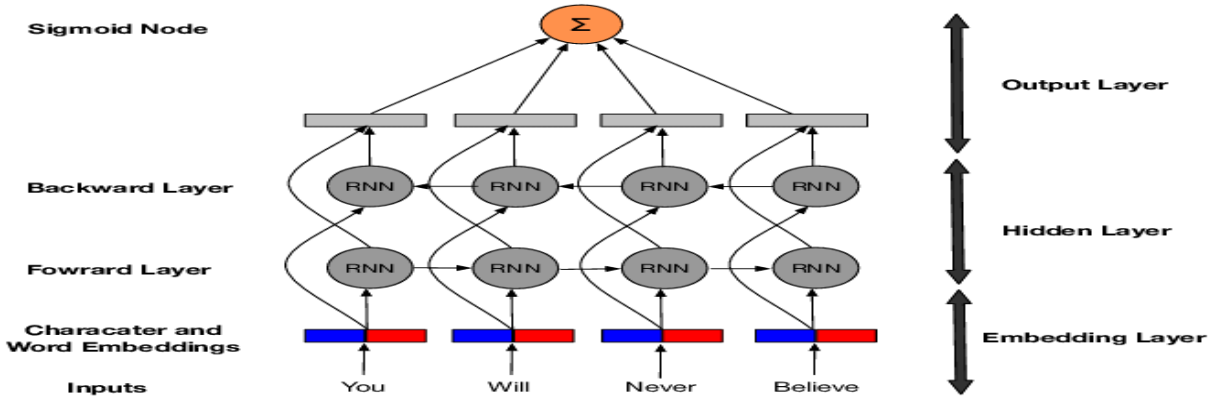


Figure 3.7: Bidirectional RNN Architecture with forward and backward hidden RNN-layers.

3.6 Long short term memory (LSTM) :

Long short-term memory (LSTM) networks [39] are an RNN extension designed to work on sequence problems by being capable of learning long-term dependencies and that has achieved state-of-the-art results on challenging prediction tasks. LSTM networks employ recurrent connections and add memory blocks in their recurrent hidden layers. These memory blocks save the current temporal state during training and make possible to learn temporal observations hidden in the input data. The fact of using connections as a memory implies that the output of a LSTM network depends on the entire history of the training data, not only on the current input sample. Moreover, using memory blocks allows relating the current data that is being processed with the data processed long before, solving the problem experienced by common RNNs. For this reason, LSTM networks have had a positive impact on sequence prediction tasks. As stated for RNN, a bidirectional layer using two hidden LSTMs can be leveraged to process data both forward and backward.

A LSTM has three gates and a memory cell state from which information can flow from one module to another with either some linear interactions or unchanged. The gates are means to remove or add information to the cell states, and they consist of a multiplication operation and a sigmoid layer. The three gates in a LSTM module are input, forget and output gates:

- **Forget Gate:** decides what portion of the previous data to discard.
- **Input Gate**: conjectures what new knowledge is to be updated in the current module.
- **Output Gate:** reflects output of the present module. [35] [37]

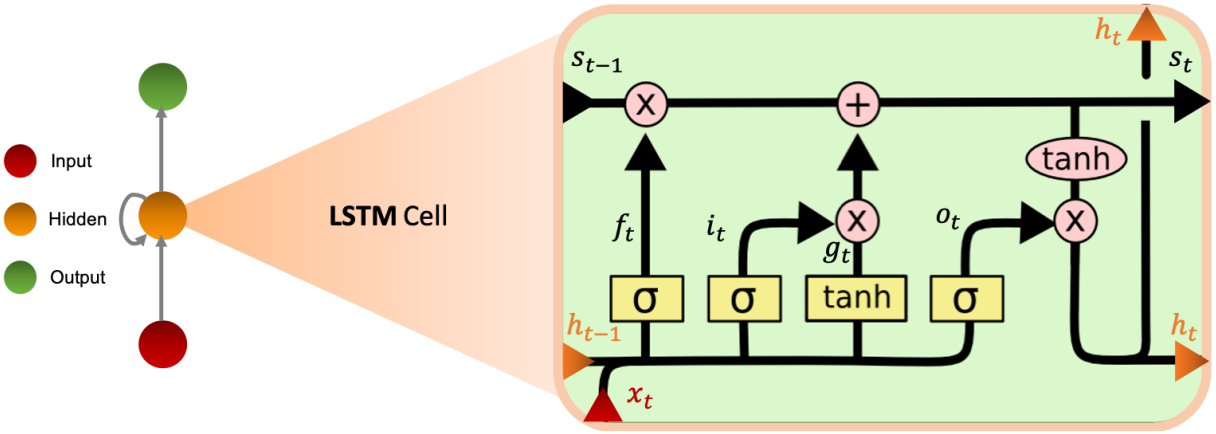


Figure 3.8: Basic Architecture of LSTM.

3.7 Learning Process of Deep Neural Networks

The training of a neural network is about learning the values of its parameters (weights w_{ij} and b_j biases) and it is performed by an iterative process of "*going and return*" by the layers of neurons. The "*going*" is a forward propagation of the information and the "return" is a backpropagation of that information. As the model is being trained, the weights of the interconnections of the neurons will gradually be adjusted until good/bad predictions are obtained. This process is repeated, layer by layer, until all the neurons in the network have received a loss signal that describes their relative contribution to the total loss.

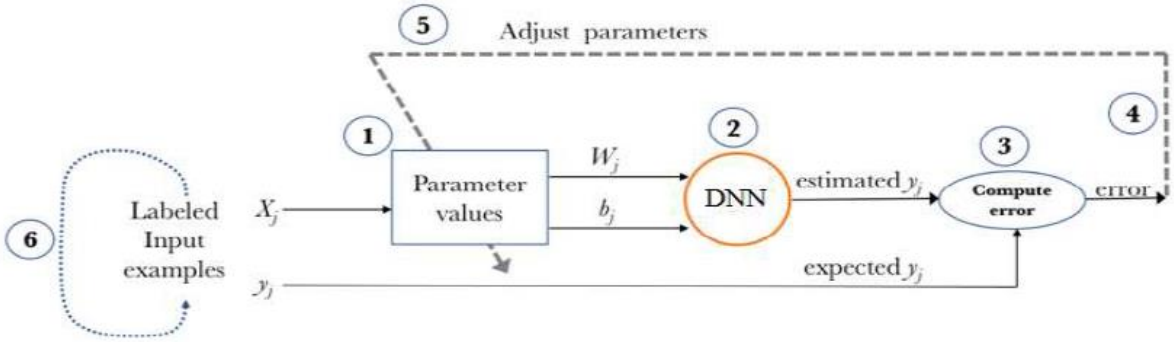


Figure 3.9: Illustration of deep neural network learning process [40].

In Figure above, numbers refer to the following steps:

1. Start with values (often random) for the network parameters (*wij* weights and *bj* biases).

2. Take a set of examples of input data and pass them through the network to obtain their prediction.

3. Compare these predictions obtained with the values of expected labels and calculate the loss with them.

4. Perform the backpropagation in order to propagate this loss to each and every one of the parameters that make up the model of the neural network.

5. Use this propagated information to update the parameters of the neural network with the gradient descent in a way that the total loss is reduced and a better model is obtained.

6. Continue iterating in the previous steps obtaining a good model [40].

3.7.1 Activation Functions:

1) Linear: The linear activation function is basically the identity function in which, the inputs are multiplied by the weights for each neuron for creating an output. This kind of activations does not support backpropagation.

2) Sigmoid: it is a nonlinear activation that allows a reduction in extreme or atypical values in valid data without eliminating them: it converts independent variables of almost infinite range into simple probabilities between 0 and 1. Most of its output will be very close to the extremes of 0 or 1.

3) Tanh: Unlike the sigmoid function, the normalized range of tanh is between -1 and 1, which is the input that goes well with some neural networks. The advantage of tanh is that negative numbers can be dealt with more easily.

4) SoftMax: it is often found in the output layer of a neural network and returns the probability distribution over mutually exclusive output classes.

5) ReLU: The activation function rectified linear unit (ReLU) is a very interesting transformation that activates a single node if the input is above a certain threshold. The

default and more usual behavior is that, as long as the input has a value below zero, the output will be zero but, when the input rises above, the output is a linear relationship with the input variable of the form *f(x)=x*. The ReLU activation function has proven to work in many different situations and is currently widely used .[40]

3.7.2 Loss Functions

1) Cross-entropy Loss: or logistic loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. A perfect model would have a log loss of 0.

Mathematically, it can be written as it is shown in formula (1):

$$\boldsymbol{CE} = -\Sigma\, \boldsymbol{tilo}(\boldsymbol{si})\, \boldsymbol{Ci} \quad \textbf{(1)}$$

Where $\boldsymbol{ti}$ and $\boldsymbol{si}$ are respectively the ground-truth and the model score of a sample for the class *i* (Gómez, 2018) .

2) Categorical Cross-entropy Loss:

Also called softmax loss. It is softmax activation plus cross-entropy loss and it is used for multi-class classification where the labels are one-hot. Its formula is written like (2):

$$\boldsymbol{CCE} = -\boldsymbol{log}\,(\boldsymbol{esp}\,\Sigma\,\boldsymbol{esj}\,\boldsymbol{Cj}) \textbf{ (2)}$$

Where $\boldsymbol{sp}$ is the model score for the positive class of the given sample [41].

3) Focal Loss:

It was introduced by [42]from Facebook the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training in dense object detection.

It is a cross-entropy loss that weighs the contribution of each sample to the loss based in the classification error.

In other words, if a sample is already classified correctly by the classifier, its contribution to the loss decreases to make the loss implicitly focus in those problematic classes. In addition, the authors also weight the contribution of each class to the loss in a more explicit class balancing.

They added to cross entropy $(\mathbf{1} - \boldsymbol{Pt})\ \boldsymbol{\gamma}$ as a modulating factor to reduce the influence of correctly classified samples in the loss and with tunable focusing parameter $\boldsymbol{\gamma} \geq \mathbf{0}$ they defined it as:

$$\boldsymbol{FL} = -\boldsymbol{\alpha t}\ (\mathbf{1} - \boldsymbol{Pt})\ \boldsymbol{\gamma log}(\boldsymbol{Pt}) \qquad (\mathbf{3})$$

Where $\boldsymbol{y}$ is the true label of the sample, $\boldsymbol{p}$ is the model prediction; the factor $\boldsymbol{lo}(\boldsymbol{Pt})$ is the rewritten CE by defining $\boldsymbol{Pt}$ as in the formula (4) below:

$$P_t = \begin{cases} p & ,if\ y = 1 \\ 1 - p & ,otherwise \end{cases} \qquad (4)$$

$\boldsymbol{\alpha t}$ is weighting factor used for addressing class imbalance with $\boldsymbol{\alpha} \in [\mathbf{0}, \mathbf{1}]$ and:

$$\alpha_t = \begin{cases} \alpha & ,if\ y = 1 \\ 1 - \alpha & ,otherwise \end{cases} \qquad (5)$$

3.7.3. Gradient Descent

The *gradient descent* is the basis of many optimizers and one of the most common optimization algorithms in machine learning and deep learning. It uses the first derivative (gradient) of the loss function when updating the parameters. The frequency of parameters updating is parameterized, so, it can be after:

- Each entry examples.
- Each round of the whole set of training examples (epoch).
- A sample of examples of the training set (batch) [40].

3.8 Machine Translation and Summarization

One way to look at summarization is to consider it as a machine translation problem: we want to translate the original text into a shorter version of this text. The idea is to use work done in machine translation, area which is very active and to adapt it to text summarization. Another criterion which links machine translation and summarization is the use of the same metrics to evaluate to results: ROUGE and BLEU scores.

3.9 Attention Layer

Attention Mechanism was firstly introduced for machine translation by [1] and then developed by [2]. Its purpose is to determine the importance of each hidden state and provide a weighted sum of all the features fed as input. This makes possible to solve traditional limits in various NLP tasks. For instance, traditional word vectors presume that a word meaning is relatively stable across sentences. However, a single word could have different meanings in different contexts, e.g., lit (*an adjective that describes something burning*) and lit (*an abbreviation for literature*); or get (a verb for obtaining) and get *(an animal's offspring*). ALs can capture the shades of meaning for a given word that only emerge due to its situation in a passage and its inter-relations with other words. Moreover, ALs learn how to relate an input sequence to the output of the model in order to pay selective attention on more relevant input features. By this way, Attention mechanism makes models able to replicate the human brain in focusing on specific parts of the input which enables them determining the importance of each part for the desired output. [35] [44]For more details of how this attention layer works, we take as example the attention layer used in [1]work and that used in the Transformer architecture of [2]for machine translation.

3.9.1. Attention Mechanism

The *Attention mechanism* is a lego block that can be used in any deep learning model that one builds.

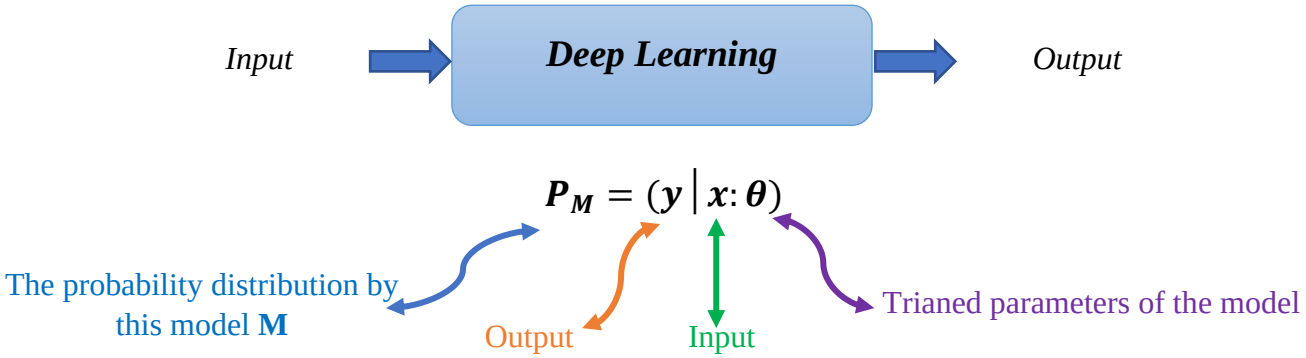


Figure 3.10: Deep learning model.

In December 2017, when the paper *Attention is All You Need* [2] was released, it was shown that an attention mechanism is apparently the only lego block you need. By this, we mean we can do away with more a traditional RNN's. These lego blocks can be entirely replaced by attention when dealing with sequential data like in our model.

In fact, the attention mechanism is so powerful that it allows models to not only perform better but also train faster, which in our world of finite computational resources is a great gift so that we can all actually benefit from these magical lego blocks.

We have sequential data each data point in our sequence is a vector of numbers, this for example word embeddings in an NLP task.

we called this input vectors **values** So we have V_1 to V_N where N we do not actually know the length of the sequence, so we are going to use some unknown variable N

[2] https://arxiv.org/pdf/1706.03762.pdf

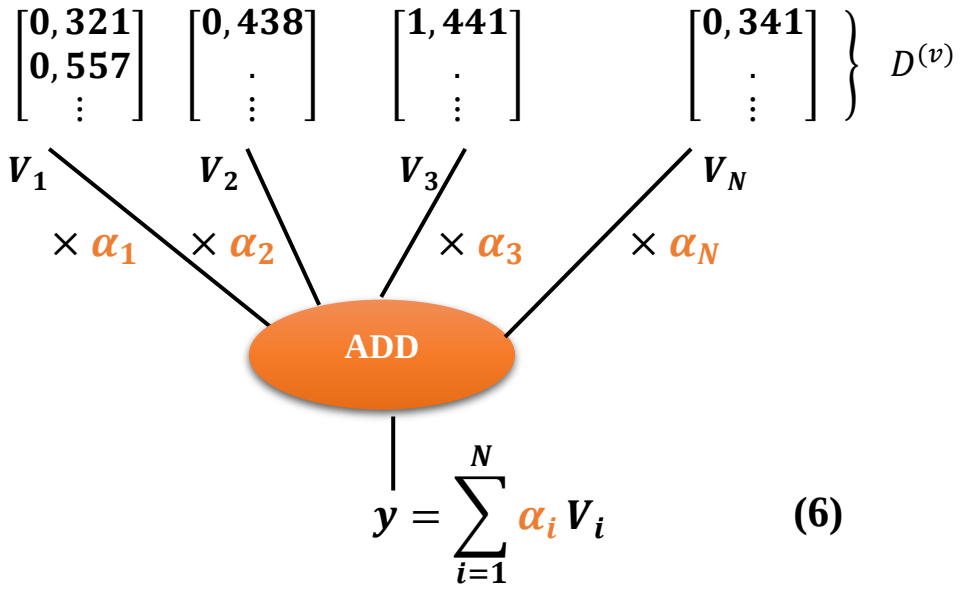


Figure 3.11: Attention mechanism illustration.

we are however fortunate in that the vector are the fixed dimension which we can call D_N,The aim of attention mechanism is to simply perform a linearly weighted sum of the value vectors .we would ideally want to train the model to learn the optimal weights the $\boldsymbol{\alpha}$ to perform best the task the model is being useful.

However we cannot simply learn the optimal $\boldsymbol{\alpha}$ because the input is variable length N can change from input sequence to the next so we would not even have fixed numbers of $\boldsymbol{\alpha}$ to train another restriction on the $\boldsymbol{\alpha}$ weights is that we do not want to scale the input vectors in any manner. **i.e** we need our $\boldsymbol{\alpha}$ to normalized which means they must sum to 1.

$$\sum_{i=1}^{N} \alpha_i = 1 \qquad (7)$$

Thus, to obtain these $\boldsymbol{\alpha}$ weights we cleverly make something called a ***key*** and ***query*** like in dictionary in python every value has an associated key often the is the value itself and this called *self-attention* however we will talk about attention generically so let's have our sequence of key vectors from $\boldsymbol{K_1}$ to $\boldsymbol{K_N}$.

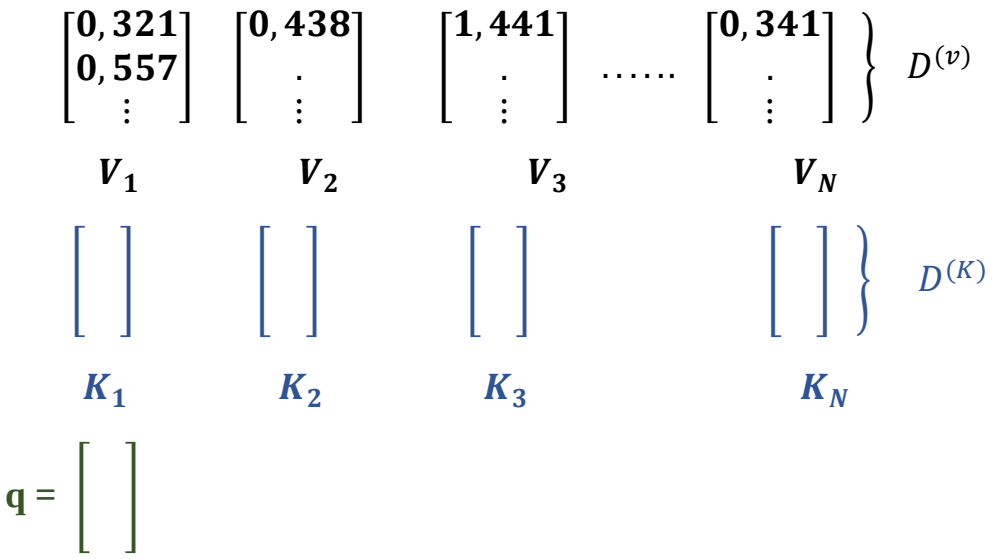


Figure 3.12: Attention mechanism illustration.

The query **q** is also a vector but for a single particular attention mechanism there will only be one query depending on the task the query can be constructed from a range of different places ,Now that we have query and key we can obtain the weightings for the values $\boldsymbol{\alpha}$ there will be some functions that combines the query and the respective key to obtain the linear weightings for the corresponding value.

$$\alpha_i = f(q, K_i) \qquad \textbf{(8)}$$

As an example a popular example is dot product attention where we simply perform the dot product between the key and query apply a non-linearity often the hyperbolic $tanh$ function and then apply the $softmax$ function to ensure the $\boldsymbol{\alpha}$ still normalized and sum to 1.

$$\{\alpha_i\}_{i=1}^{N} = softmax(\{tanh(q^T K_i)\}_{i=1}^{N}) \qquad \textbf{(9)}$$

We summarize the attention mechanism in the figure below.

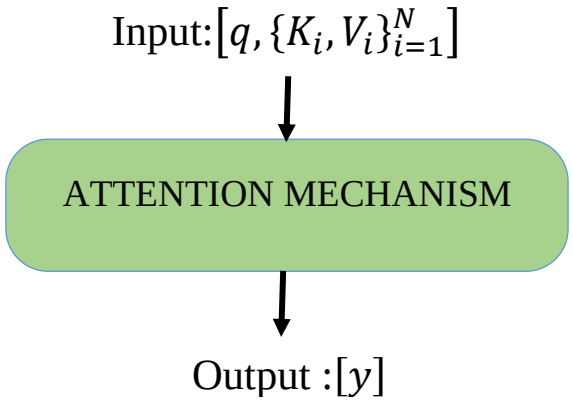


Figure 3.13: Attention mechanism illustration.

All the equations here are very deterministic we do not have any learnable parameters that can be updated during the training of that model uses attention we can introduce fixed size matrices of learnable parameters by performing linear transformations of our keys values and query.

$$\boldsymbol{q} = W^{(q)}\, q' \qquad \boldsymbol{k_i} = W^{(k)} k_i' \qquad \boldsymbol{V_i} = W^{(V)} V_i' \qquad \textbf{(10)}$$

$W^{(q)}, W^{(K)}, W^{(V)}$ the parameters that can be learnt by the model and can be trained to perform as well as possible on the task at hand by updating these parameters.

The point is to emphasize that attention mechanisms are literally just a function of inputs to outputs and, thus, can be used as part of any model in supervised and unsupervised tasks. It is a good way to start the day.

3.9.2. Attention in Translation

In the figure, the top part of the network used for LSTM-based machine translation extracts features from the first word via an encoder. The features of the next word are extracted from the next word and the features of the first word. This process is repeated to extract features from all words. The result of this processing is that the LSTM output vector, once the last word is reached, retains the features of the whole sentence. The bottom decoder is used to take the feature vector obtained by the encoder and generate a translation of the original sentence.

From the first vector, the first word is output. Then, a feature vector is generated from the first vector and the first word to produce the second word. This process continues for all words. This process is similar to that of a person who reduces the content of a sentence that they hear to a concept and then recreates a sentence based on this concept. When a human actually translates content that needs to be translated accurately, we would assume that the process involves finding a corresponding match for each word in the source language. In translating this sentence, for example, first we have the subject, which is "hamburger", then we have the Japanese for "*in this restaurant"*, which has not been translated yet. As the process continues in this way, the sentence is translated more accurately by linking each word in the source to a word in the target. We will use attention to focus on the first word that should be translated and output this first word. Based on this output, the neural network RNN determines the next word of focus in the original sentence and outputs the next word. This process is repeated to perform the translation process.

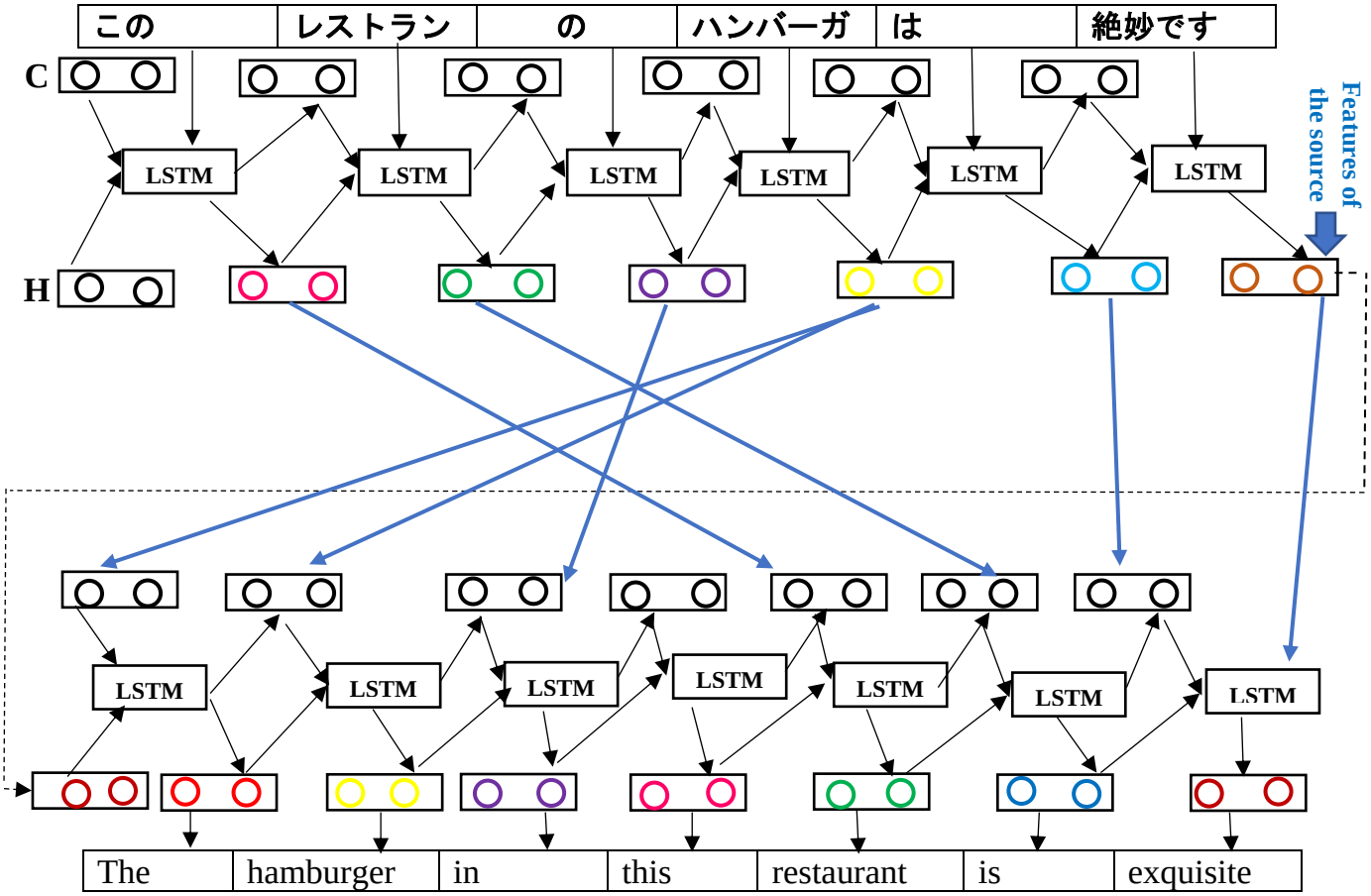


Figure 3.14: Example of language translation using LSTM.

3.9.3. Implementation of Attention

The attention configuration here in this figure will determine what part of the input sentence is in focus as the network outputs each word. To estimate the first word in the translation, the inner product of the initial vector of the decoder and the feature vector of each word, which is the output of the encoder, are calculated.

Attention produces a score for each word, whereby larger scores indicate the words of focus. The softmax calculation is performed on this score. This operation converts to a probability expression in which the sum of score values is 1.

This represents the location of focus in the input series. In this case, the word is hamburger. The softmax value is multiplied by the feature vector output by the encoder at each time step in the process. The feature vector for each word is added to each dimension in order words. A weighted sum operation is performed on the feature vector so that words with a higher degree of focus have greater impact. The first word of output is derived based on the combined features and the LSTM output.

This process repeats to drive the next word in the sentence. The next word is determined by calculating the inner product of the output LSTM for the previous time step and the features at each time step of input and then obtaining the softmax value. In this case, it is still the word "hamburger". A weighted sum operation is performed on the feature vector using the softmax weight and then this is concatenated with the LSTM output to output the next word. This process repeats until the entire sentences are output.

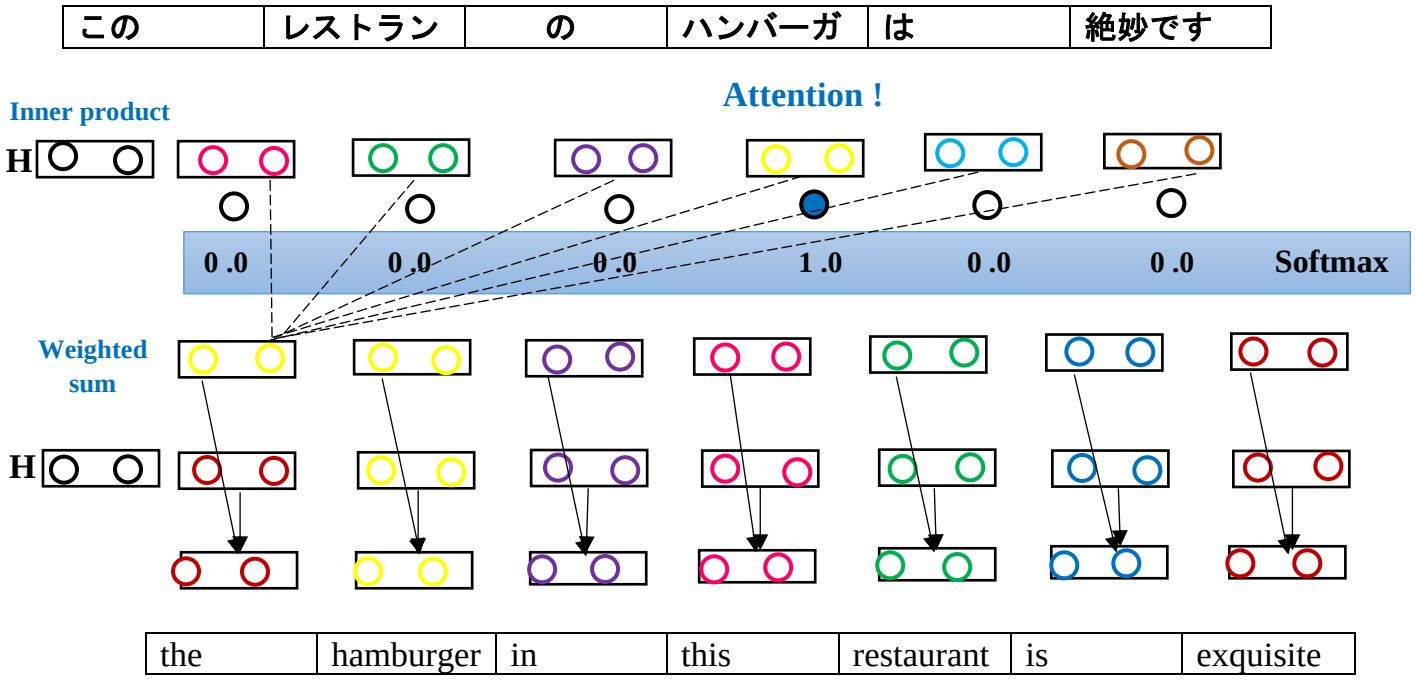


Figure 3.15: Example of language translation using LSTM and attention.

3.10 Embeddings

Embedding is a mathematical method of mapping an object in one domain to another object in another domain. Embedding is widely used as a tool in NLP tasks. There are supervised and unsupervised ways to generate embeddings. In this thesis, we are more interested in unsupervised embeddings. Unsupervised embeddings tend to be more generic and have a wider range of uses.

3.10.1 Word Embeddings

Word embeddings map a word to a multidimensional vector. Most of these mapping methods build upon vector space model, saying an object can be described by its relations to other objects. Word embeddings have been in use since 60s. However, they became popular with word2vec paper by Mikolov et al [30]. Word2vec allowed researchers to use huge datasets to train word embeddings in a short time. The driving idea behind word embeddings is, words can be encoded using other words they are used together.

Word2vec uses two methods for such use: first is CBOW [29], and the second is skip grams [30]. In CBOW method, current word is predicted using neighboring words. In skipgram method, neighboring words are predicted using current word.
In 2017, Bojanowski et al released a word embedding solution named fasttext [32] Fasttext introduced sub word embeddings that enabled it to generate an embedding even when it did not see that word before.

3.10.2 Sentence Embeddings

Sentence embeddings turn a sentence into a vector. They are useful in comparing sentences to each other, classifying sentiment or answering questions. These embeddings can be built upon words or characters [31]. Sentence embeddings can be built upon word embeddings, or they can be generated using source document and words in the sentence. Sentence embeddings provide a middle ground between abstraction of the sentence and the performance.

There are some advantages of using sentence embeddings over word embeddings. First, sentence embeddings can deal with polysemy better than word embeddings. A word embedding solution does not have access to information needed to distinguish between different meanings of a word, but sentence embeddings can use the whole sentence to figure out which meaning or use was intended. Second advantage is, when using word embeddings, word order has to be processed by our solution. When using sentence embeddings, word order is processed by sentence embedding solution.
Most word embedding solutions use CBOW or skip grams and these methods do not capture word order at all.

Furthermore, sentence embeddings can be trained on very large corpora. This way, sentence embeddings can learn relationships that are not easily seen in our documents. These relationships can help in our final task. One can use a variety of ways to generate sentence embeddings. The most basic way of creating sentence embeddings involves using a vector space model. We can model each sentence using the words it has. Then, we can assign each word an index and set those indices to one when that particular word is used in the sentence. Number of different words possible poses a challenge, but we can just limit the number of words we include in our dictionary.

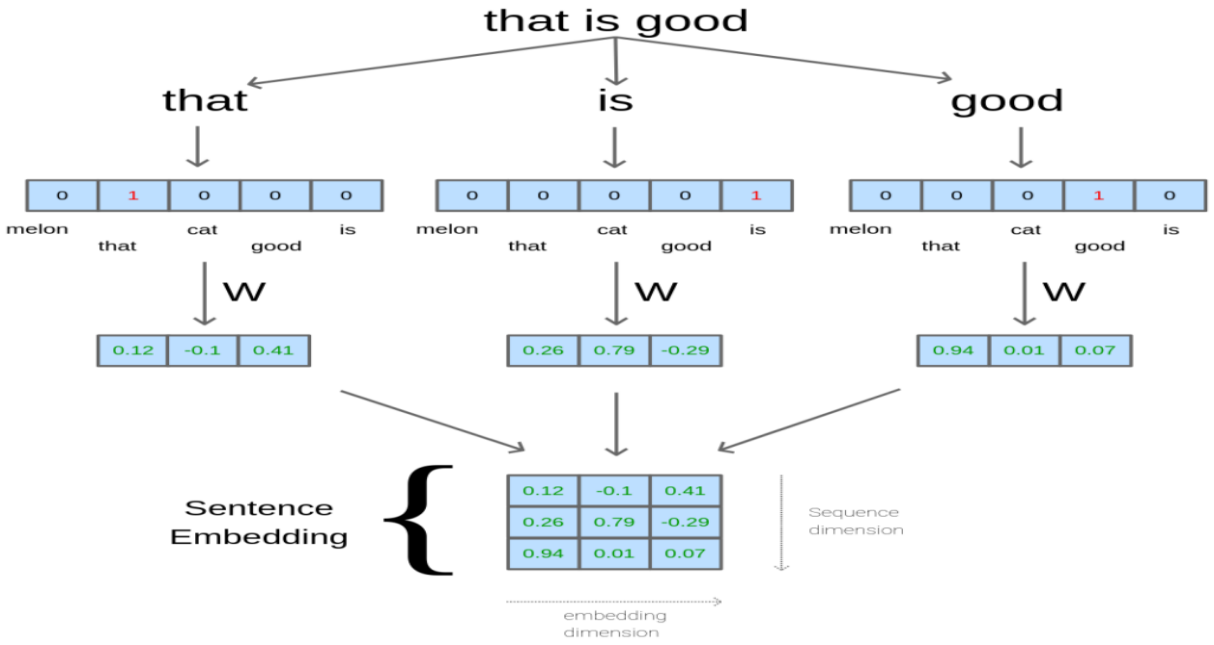


Figure 3. 16: Sentence embedding illustration.

3.11 Dense layer

A dense layer is a densely connected layer that is used to map large unit inputs in a few unit results. For example, it may be used to define the number of classes that a model returns, mapping hundred and thousand nodes in a few numbers of classes [35].

3.12 Proposed Approach Steps

In Figure 3.7 below, illustrates the process of generating Neural abstractive summary

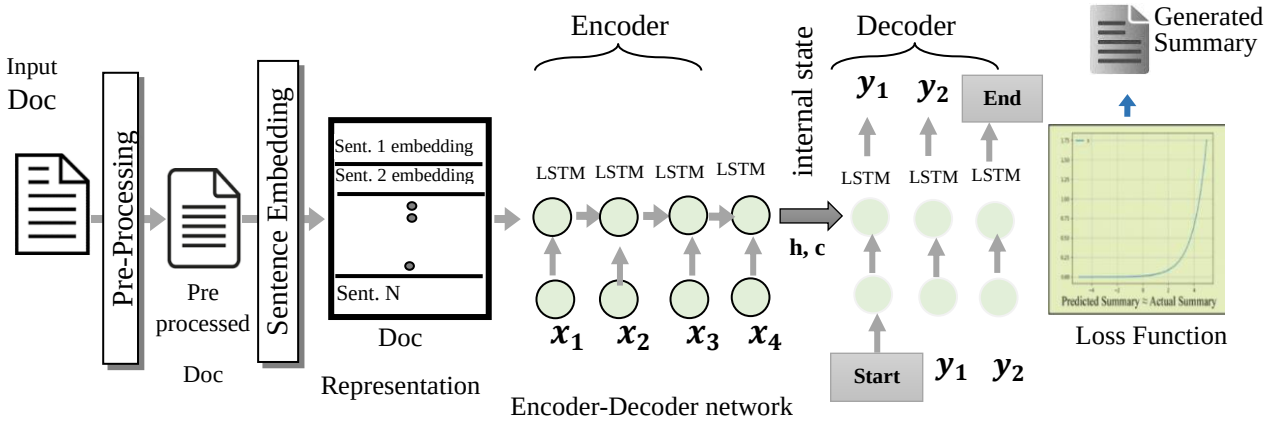


Figure 3.17:Illustration of Proposed Abstractive summarization Framework.

3.12.1 Pre-processing

Performing basic preprocessing steps is very important before we get to the model building part. Using messy and uncleaned text data is a potentially disastrous move. Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing *is a technique that is used to convert the raw data into a clean data set.* We followed the steps below:

- Convert everything to lowercase.
- Remove HTML tags.
- Contraction mapping.
- Remove (‘s).
- Remove any text inside the parenthesis ().
- Eliminate punctuations and special characters.
- Remove stopwords.
- Remove short words (length (word)=1).

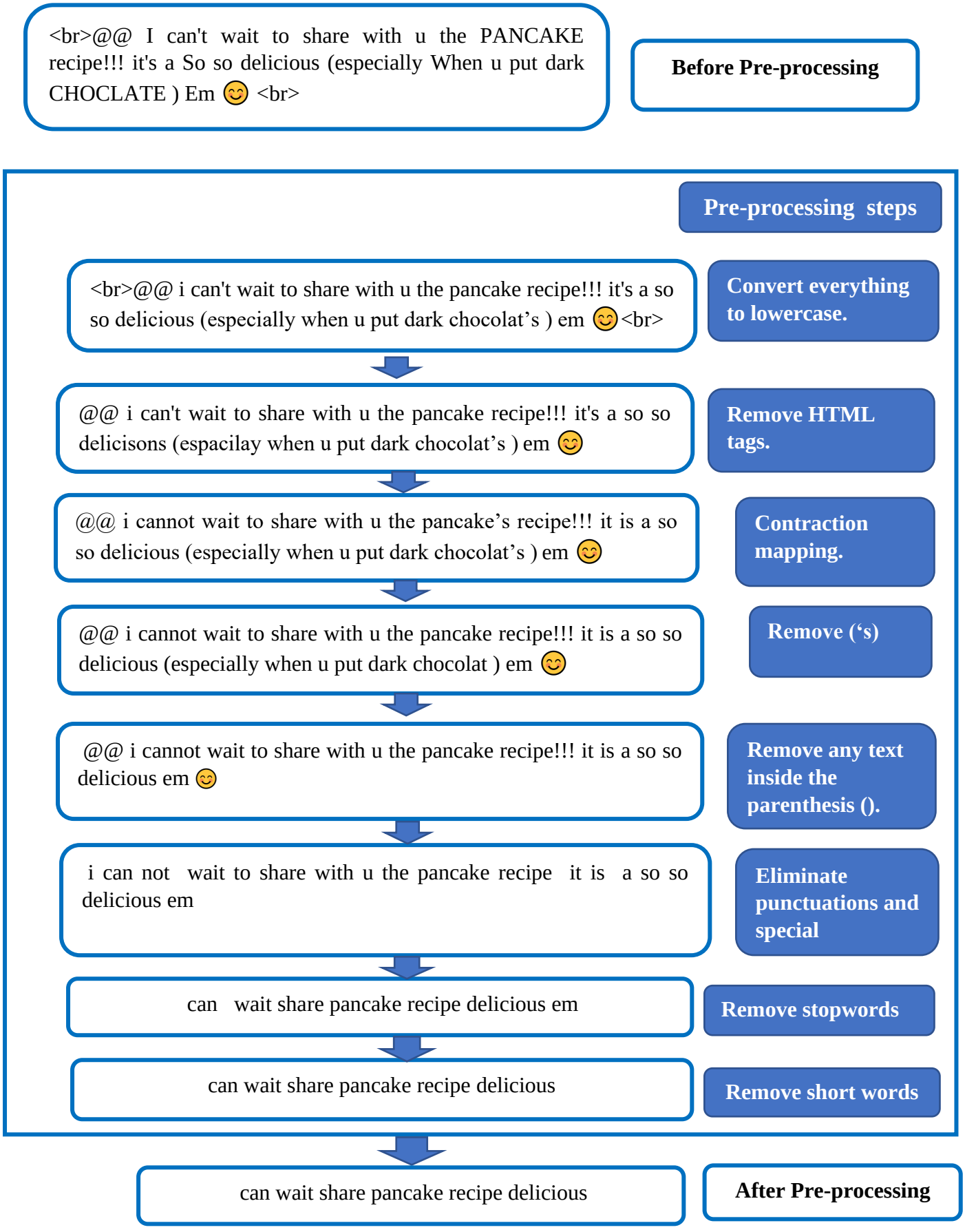


Figure 3.18: Pre-processing procedure illustration.

Because of our model, we needed to transfer our original sentence to a civilization sentence and transfer the summary to a one-hot matrix. As we see in the figure below:

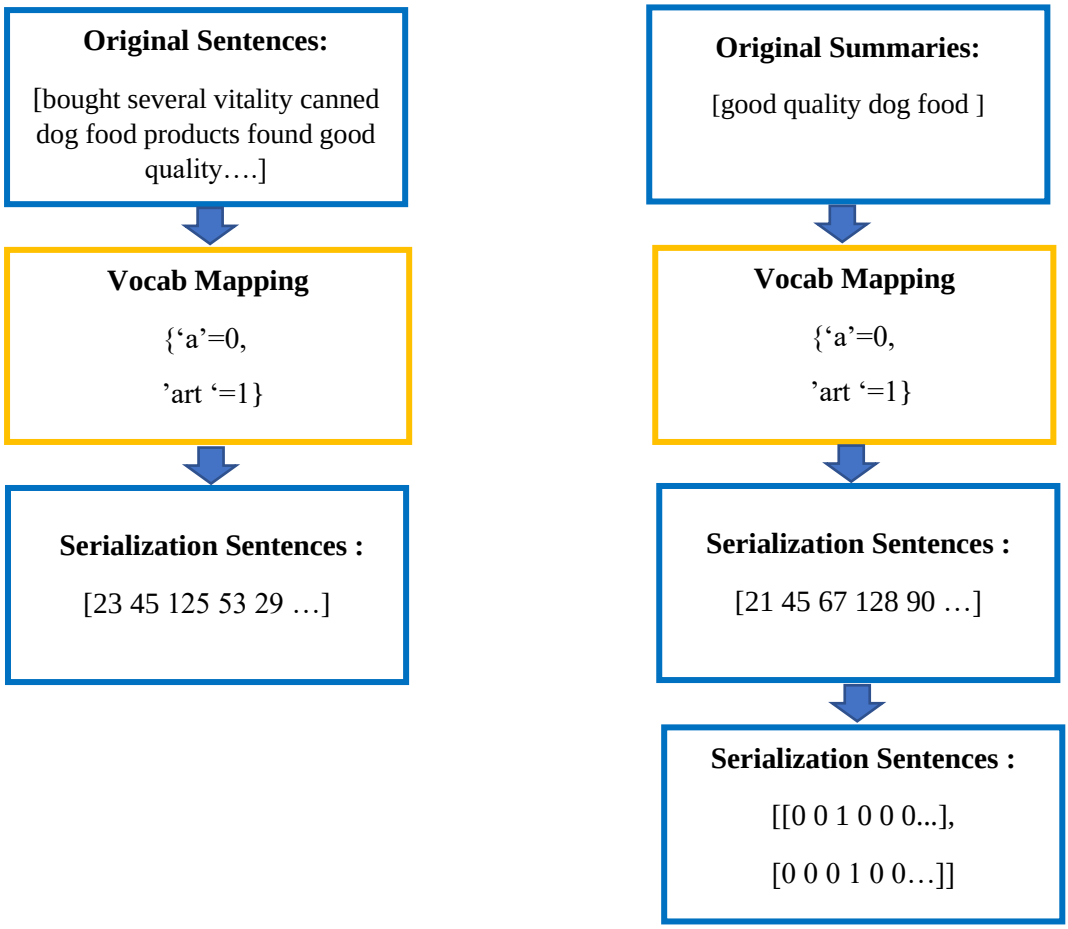


Figure 3.19: Abstractive summarization preprocessing.

3.12.2 Attention in abstractive summarization

We will introduce the attention model of historical summarization. The main improvement to the attention model is the context vector. The context factoring attention model is derived from the hidden layer state of the encoder, so it will capture or garb more information from the input text.

The attention layer has two inputs **a** and ***s_prev*** . **a** is the hidden layer satete of the encoder. The ***s_prev*** is the last round of the hidden layer output of the decoder. The first step is to concatenate S and edage in the figure. The next step is to encode the concat third factor into our model. Notice the dealer model has a softmax output layer so we can get corresponding

weights of the hidden layer state of the encoder and finally just states the waste weight so we can get the context vector.

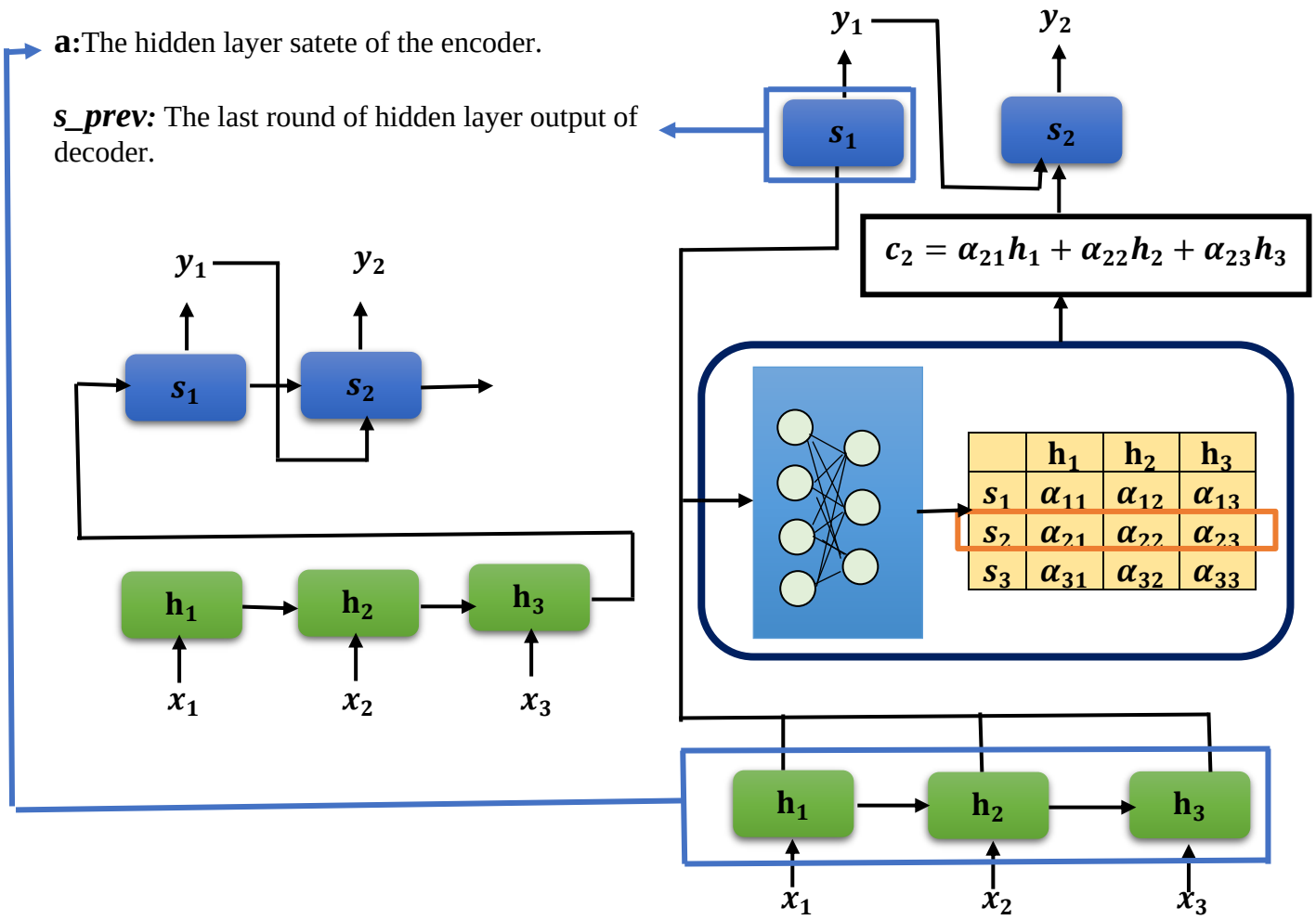


Figure 3.20: Abstractive summarization attention implementation.

3.12.3 Sequence-to-Sequence (Seq2Seq) Modeling

A sequence-to-sequence model consists of an encoder and a decoder. The encoder is responsible for reading the input tokens. The input tokens represented by their vectors known as word embeddings are read and fed into a bi-directional LSTM to generate the encoder's

hidden states. The decoder receives the final encoder state and feeds it along with its input to a uni-directional LSTM to generate the decoder's hidden state for the given time step.

To help decoders generate the correct outputs, the distribution of the input tokens known as attention is calculated. Next, the context vector, the weighted average of attention weights, is calculated and the decoder uses this weight to generate the final distribution over all the words in the vocabulary.

Finally, the word with the maximum probability is selected as the output of that time step. Our work is clearly aligned with recent studies on sequence-to-sequence models [33] for abstractive summarization. A neural attention-based abstractive summarization model [14] for headline generation inspired by [34] on short texts (one or two sentences) is one of the earliest attempts to use neural networks to beat the performance of the existing traditional summarization models. The work done by Chopra et al. [15] is an extension of [14] which computes the conditional probabilities of next words using a recurrent decoder, improving the results of short text summary generation. An attentional encoder-decoder recurrent neural network for abstractive summarization was introduced by [17].

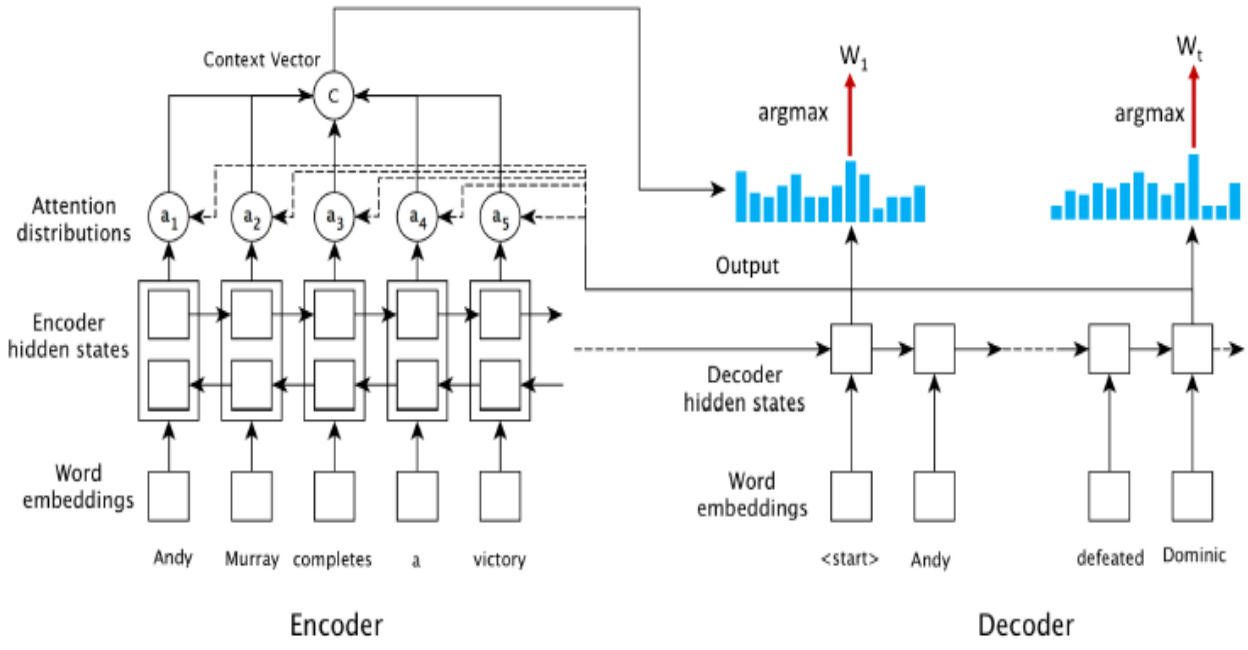


Figure 3.21: Sequence-to-sequence model with attention mechanism.

3.12.4 Prediction

In our prediction, after we get the model output, we can have a probability matrix which is the softmax output. Our goal is to get the most likely summary, so we need to decode it by arcmax or beam search and map the number to a word, so we can get our predicted summary.

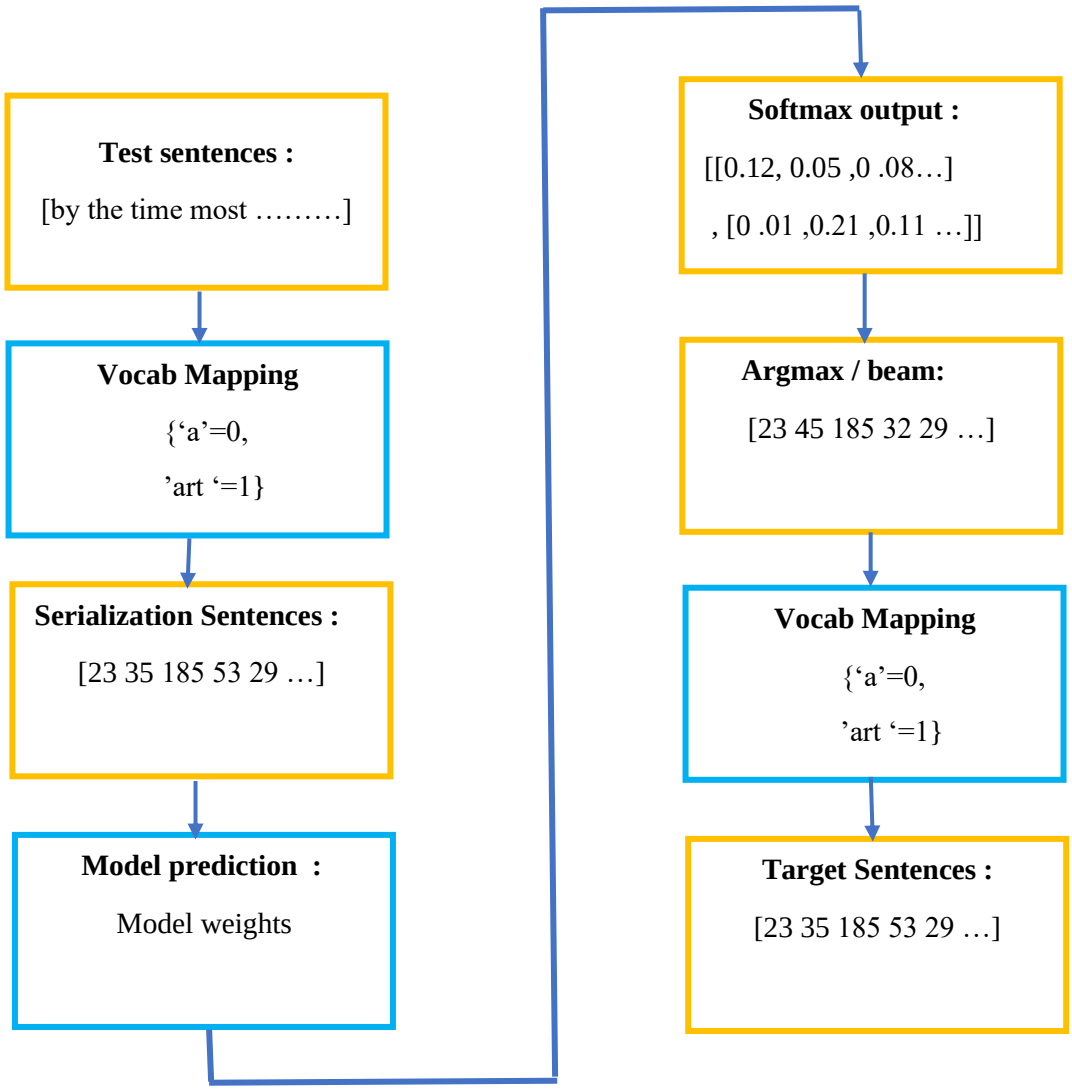


Figure 3.22: Abstractive summarization prediction illustration.

3.13 Architecture

The idea behind the design of this model is to enable it to process input where we do not constrain the length. One RNN will be used as an encoder, and another as a decoder. The output vector generated by the encoder and the input vector given to the decoder will have a fixed size. However, they need not be equal. The output generated by the encoder can either be given as a whole chunk or can be connected to the hidden units of the decoder unit at every time step.

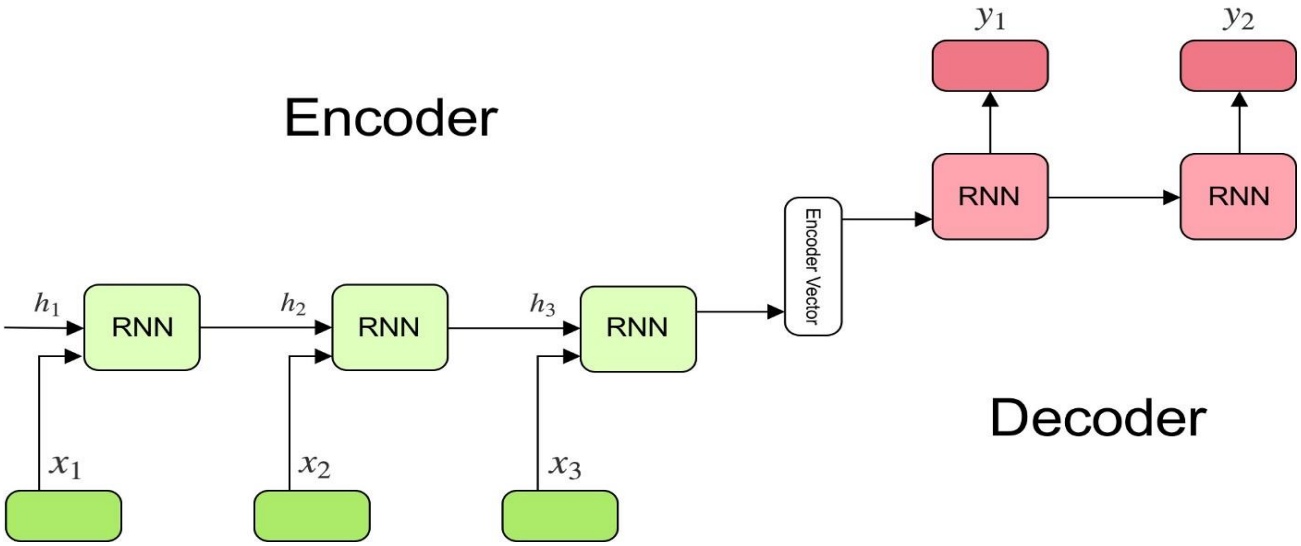


Figure 3.23: Encoder-Decoder Sequence-to-Sequence.

Text Summarization Using an Encoder-Decoder Sequence-to-Sequence Model

- Step 1 - Importing the Dataset.
- Step 2 - Cleaning the Data.
- Step 3 - Determining the Maximum Permissible Sequence Lengths.
- Step 4 - Selecting Plausible Texts and Summaries.
- Step 5 - Tokenizing the Text.
- Step 6 - Removing Empty Text and Summaries.

3.13.1 Encoder

The input length that the encoder accepts is equal to the maximum text length which you've already estimated in Step 3. This is then given to an Embedding Layer of dimension (total number of words captured in the text vocabulary) x (number of nodes in an embedding layer)

(calculated in Step 5; the x_voc variable). This is followed by three LSTM networks wherein each layer returns the LSTM output, as well as the hidden and cell states observed at the previous time steps.

3.13.2 Decoder

In the decoder, an embedding layer is defined followed by an LSTM network. The initial state of the LSTM network is the last hidden and cell states taken from the encoder. The output of the LSTM is given to a dense layer wrapped in a time-distributed layer with an attached *softmax* activation function.

The model accepts encoder (Text) and decoder (Summary) as input and outputs the summary. The prediction happens by predicting the upcoming word of the summary from the previous word of the summary (see the below figure).

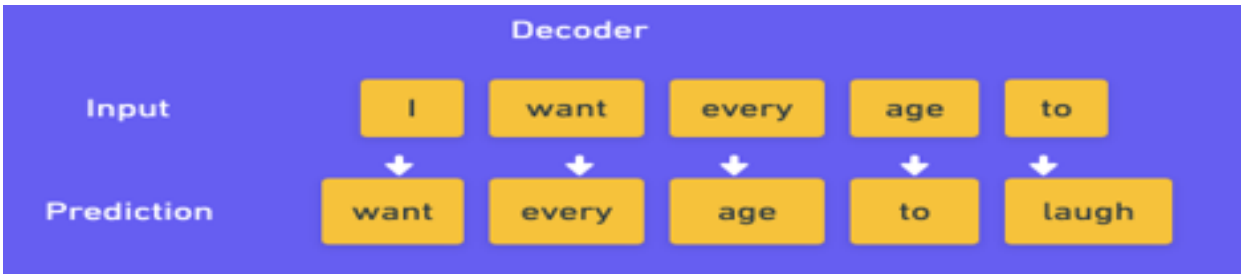


Figure 3.24: Encoder-Decoder**.**

Note: Consider the summary line to be "*I want every age to laugh*". The model has to accept two inputs - the actual text and the summary. During the training phase, the decoder accepts the input summary given to the model, and learns every word that has to follow a certain given word. It then generates the predictions using an inference model during the test phase.

3.14 Abstractive cross-language summarization (Translate and Summarize)

According to Junnan Zhu et al « Cross-lingual summarization is to produce a summary in a target language (e.g., English) from a document in a different source language (e.g., Chinese). Cross-lingual summarization can help people efficiently understand the gist of an article written in an unfamiliar foreign language. » [33]

In our model, we decided to use different languages. (Arabic, French, English) to see more results. First, the input is two different languages, as it appears in the figure below:

In this example, we translated a part of our data to Arabic, then we applied our model and the result was so satisfying. We will see that in chapter 4.

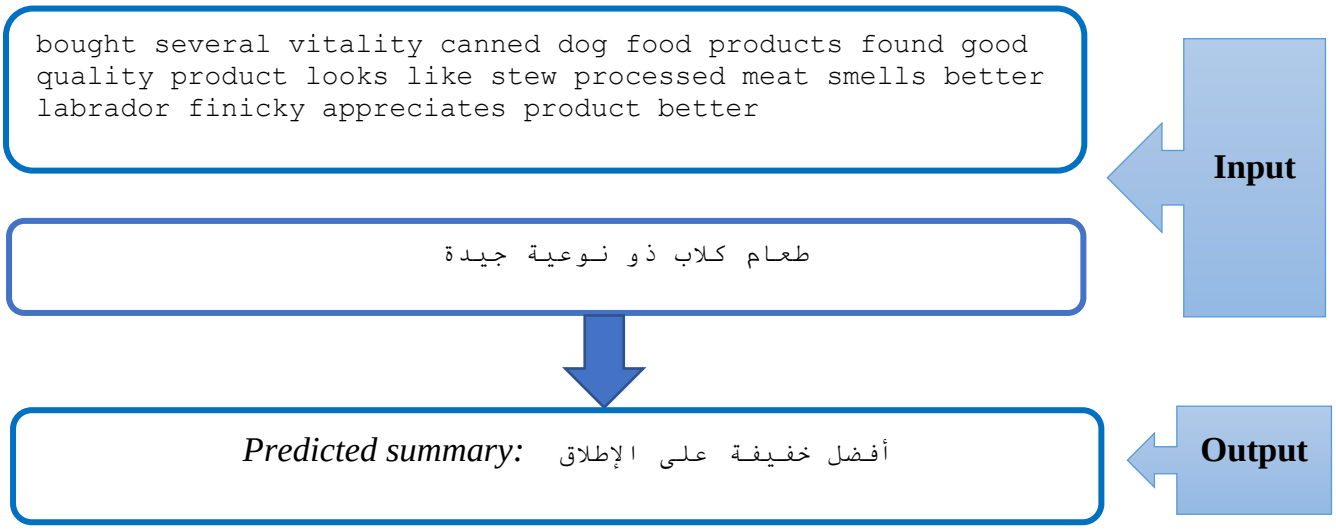


Figure 3.25: Abstractive summarization (English-Arabic).

Secondly, the input is one language and the output is another language, like we see in the figure.

In this case, we translated the input and the output. The input was translated from English to Arabic and the output from Arabic to French. The result was very good.

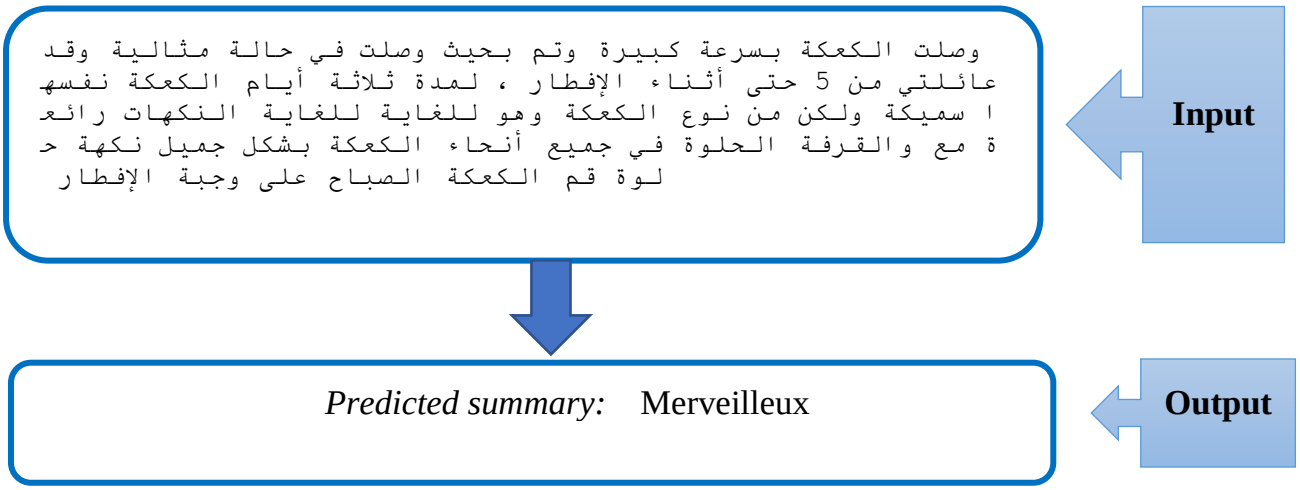


Figure 3.26: Abstractive summarization (Arabic-French).

3.15 Conclusion

This chapter discussed the research framework and, in general, the methodology required to implement the four decision problems according to the research questions formulated in Chapter 1. We presented five phases, namely analysis, design, development, implementation, and evaluation; each plays an important role in accomplishing the objectives of this research. While we discussed the methodology designed for the four problems, and we applied cross abstractive summarization CLA to our model using three different languages, the following chapter discusses the results we obtained from the qualitative study.

Chapter 4

Test and Evaluation

4.1 Introduction

In abstractive summarization, we focus on improving abstractive methods because of their ability to generate novel sentences. Knowing what the existing summarization systems are able to offer and what issues they are facing can help future design decisions be made much more wisely.

In this chapter, we provide the results of our proposed approach measured on datasets and on the translated datasets to different kinds and languages.

4.2 Environment and Development Tools

4.2.1 Material

We used a computer DELL processor (Intel(R) Core (TM) i5-4310U CPU @ 2.00GHz 2.60 GHz) and RAM memory installed (8,00 Go)

- **Session Time:** jupyter provides 19 hours for each model developed in this search of continuous execution for one session.

4.2.2 Python Modules Used

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. The simplicity and easy to learn Python syntax emphasize readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity

and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.[3]

- **For Deep Learning Implementation**

Tensorflow[4]: is an open-source library of software for dataflow and differential programming for various tasks. Similarly, TensorFlow is used in machine learning by neural networks. Developed by Google in 2011 under the name DistBelief, TensorFlow was officially released in 2017 for free. The library is able to run on multiple CPUs and GPUs and is available across multiple platforms, including mobile. The name comes from multidimensional arrays known as tensors, which are commonly used in neural networks.[5]

Keras[6]: is an open-source library of neural network components written in Python. Keras is capable of running atop TensorFlow, Theano, PlaidML and others. The library was developed to be modular and user-friendly; however, it initially began as part of a research project for the Open-ended Neuro-Electronic Intelligent Operating System or ONEIROS. The principal author of Keras is Francois Chollet, a Google engineer who also wrote XCeption, a deep neural network model. While Keras officially launched, it was not integrated into Google's TensorFlow core library until 2017. Additional support has also been added for Keras integration with Microsoft Cognitive Toolkit.[7]

- **For Textual Data Pre-processing and Preparation**

Pandas[8]: is an open source, BSD-licensed library providing high-performance, and easy-to-use data structures and data analysis tools for the Python programming language. The use of this module eases the data sheets reading (. e.g. csv and excel files), data frame creation, organization and manipulation.

We used Pandas to read datasets csv files and their data frames manipulation.

Numpy[9]: is a python module distributed under a liberal BSD license which means that is a free module and its code is publicly available. Numpy offers comprehensive mathematical

[3] https://www.python.org/doc/essays/blurb/
[4] https://www.tensorflow.org/
[5] https://deepai.org/machine-learning-glossary-and-terms/tensorflow
[6] https://keras.io/
[7] https://deepai.org/machine-learning-glossary-and-terms/keras
[8] https://pandas.pydata.org/
[9] https://numpy.org/

functions, random number generators, linear algebra routines, Fourier transforms, and more. Moreover, it has a high-level syntax and supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

Googletrans[10]: Googletrans is a free and unlimited python library that implemented Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detect and translate. This fast and reliable library uses the same servers that translate.google.com uses for performing tasks like auto language detection, and translation.

In our work, we use this library for applying translation and back translation task strategy used for datasets augmentation.

Nltk[11] : nltk is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification,tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Re[12]: this module was added in Python 1.5, and provides Perl-style regular expression patterns. Earlier versions of Python came with the regex module, which provided Emacs-style patterns. This regex module was removed completely from Python 2.5. Re module in our work is used for pre-processing step to write regular expressions in order to find special characters in texts, and replace HTML, emoticons, and numbers by their related tags described in chapter 3.

Warning[13] : Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program. For example, one might want to issue a warning when a program uses an obsolete module.

[10] https://pypi.org/project/googletrans/
[11] https://www.nltk.org/
[12] https://docs.python.org/fr/2/howto/regex.html
[13] https://docs.python.org/3/library/warnings.html

tqdm[14]: tqdm is a library in Python which is used for creating Progress Meters or Progress Bars. tqdm got its name from the Arabic name taqaddum which means 'progress'. Implementing tqdm can be done effortlessly in our loops, functions or even Pandas.

beautifulsoup[15]: Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

Rouge[16] : There are rough implementations available for Python, but some are not native Python due to their dependency on the Perl script, and others provide differing results when compared with the original implementation. This makes it difficult to directly compare it with known results.

4.3 Data Used for Training and Test

All data in one sqlite database. 200000 food reviews Amazon users left up to October 2012. Brief summary of the review (195744 unique value), Text of the review 193579 unique value).

4.4 Experimental Results and Discussion

In this section, we initially presented the results obtained from our model.

This model can create relevant summaries for reviews written about fine foods sold on Amazon. Here are some examples of reviews and their generated summaries:

[14] https://pypi.org/project/tqdm/
[15] https://pypi.org/project/beautifulsoup4/
[16] https://pypi.org/project/rouge-score/

4.4.1 Results from ATS English-English

```
Review: delightful product buying pretty easy absolutely addicting everyone wants carmel corn product made packages last week
s yummy
Original summary: scrumptious
Predicted summary:  delicious


Review: loves tastes good meaty cousin eat meat miss cousin put chopped cilantro onions sour cream top good time
Original summary: love this stuff
Predicted summary:  great flavor


Review: found store carry thankful amazon com son says favorite meal
Original summary: taste great
Predicted summary:  great for the price


Review: previously using recommended pharmacist worked expensive taste good decided try tea yogi since reasonably priced work
s well tastes much better recommend trick
Original summary: works and tastes good
Predicted summary:  good stuff
```

Figure 4.1: Results from ATS English-English.

4.4.2 Results from ATS English-French

```
Review: check nutritional info sodium grams daily requirement used casserole ruined salty price box cost dollar unfortunately
bought two boxes throwing box away bargain wildly un delicious un healthy
Original summary: nasty even be sold no stars
Predicted French_summary: excellente collation faible en sodium


Review: special treat babies pricey would think dogs would like feel look like vegetable shaped crayons full sorts goodies al
though feeding grain free diet first ingredient hydrolyzed wheat protein
Original summary: my dogs love love love
Predicted French_summary: mes chiens adorent ça


Review: great way start day gives lots energy feel hungry also good way keep weight control like standard special nice twist
Original summary: great way to start the day
Predicted French_summary: bon produit
```

Figure 4.2: Results from ATS English-French.

4.4.3 Results from ATS English-Arabic

```
Review: loves tastes good meaty cousin eat meat miss cousin put chopped cilantro onions sour cream top good time
Original summary: love this stuff
Predicted Arabic_summary: نكهة رائعة


Review: found store carry thankful amazon com son says favorite meal
Original summary: taste great
Predicted Arabic_summary: أحب هذه الأشياء


Review: previously using recommended pharmacist worked expensive taste good decided try tea yogi since reasonably priced work
s well tastes much better recommend trick
Original summary: works and tastes good
Predicted Arabic_summary: منتج جيد


Review: daughter celiac really missed kraft macaroni cheese luckily found annie loves lives ireland ship case
Original summary: delicious mac cheese meal
Predicted Arabic_summary: جبنة ماك كبيرة
```

Figure 4.3: Results from ATS English-Arabic.

4.4.4 Results from ATS French-French

```
Review: j'ai acheté comme cadeau pour une avec le ils ont fonctionné parfaitement pour 25 petits sacs de bonbons
j'ai acheté 5 si vous voulez plus de 25 petits sacs un plus grand sac
Original summary: très bien
Predicted summary:  bon


Review: j'ai adoré ce produit pour les fruits et autres aliments avec je ne pas que mon nouveau bébé je l'ai aussi
utilisé pour mettre de la quand mon bébé ses dents cela a très bien fonctionné pour cela ainsi que pour la nourrit
ure je pas les bananes à nettoyer
Original summary: idéal nourriture
Predicted summary:  bon


Review: j'ai vraiment adoré la saveur épicée qu'ils avaient je me suis à aimer le bouillon plus que les ce qui est
généralement le si vous pas à la chaleur cela pourrait vous et si vous aimez les plats chauds cela pourrait ne pas
Original summary: saveur
Predicted summary:  bien

Review: même si nous sommes des nous pas été complètement déçus par ce produit nous eu aucun problème avec l'expéd
ition mais la crème elle même est plus a part ça on est content
Original summary: super
Predicted summary:  bon
```

Figure 4. 4: Results from ATS French-French.

4.4.5 Results from ATS French-Arabic

```
Review: pour une raison amazon ne les pas en alors je les ai à mes petits enfants qui ont pu en profiter ce sont d
e très bonnes frites amazon devrait l'expédition vers je ferais des en ligne
Original summary: super chips
Predicted Arabic_summary: ممتاز


Review: l'un des meilleurs jus de citron que j'ai essayé je suis un client en raison de la haute qualité de ce pro
duit lorsque la des frais est celui ci notre besoin d'agrumes dans la cuisine ou de br br je à l'acheter
Original summary: excellent citron
Predicted Arabic_summary: جيد


Review: que les cafés et du sud ont cette café en tant que de coffee nous avons que le était le de et nous sommes
j'ai déjà passé en le mélange et biologique mais je dois dire que ce café ici est une boisson parfaite pour l'aprè
s midi br j'ai des à la et ils donné une demi livre de depuis eux nous sommes vous ne serez pas déçus
Original summary: café adore
Predicted Arabic_summary: افضل


Review: nous qui votre produit se de et nous savoir que vous vous en aussi lorsque vous les de à cela le et le cha
ngement et la de votre les de vos des enfants de vos et des de nous aimons votre produit et que vous à la récolte
sans afin que nous continuer pour l'acheter que nous ne pas au et aux
Original summary: bon produit

Predicted Arabic_summary: جيد
```

Figure 4.5: Results from ATS French-Arabic.

4.4.6 Results from ATS French-English

```
Review: j'aime les manger chaque fois que j'ai besoin d'un regain d'énergie br il a tout ce que je recherche dans un br la pl
us le chocolat n'est jamais une mauvaise chose
Original summary: grand
Predicted English_summary: Well


Review: j'ai acheté comme cadeau pour une avec le ils ont fonctionné parfaitement pour 25 petits sacs de bonbons j'ai acheté
5 si vous voulez plus de 25 petits sacs un plus grand sac
Original summary: très bien
Predicted English_summary: Well


Review: j'ai adoré ce produit pour les fruits et autres aliments avec je ne pas que mon nouveau bébé je l'ai aussi utilisé po
ur mettre de la quand mon bébé ses dents cela a très bien fonctionné pour cela ainsi que pour la nourriture je pas les banane
s à nettoyer
Original summary: idéal pour la
Predicted English_summary: Well
```

Figure 4.6: Results from ATS French-English.

4.4.7 Results from ATS Arabic-Arabic

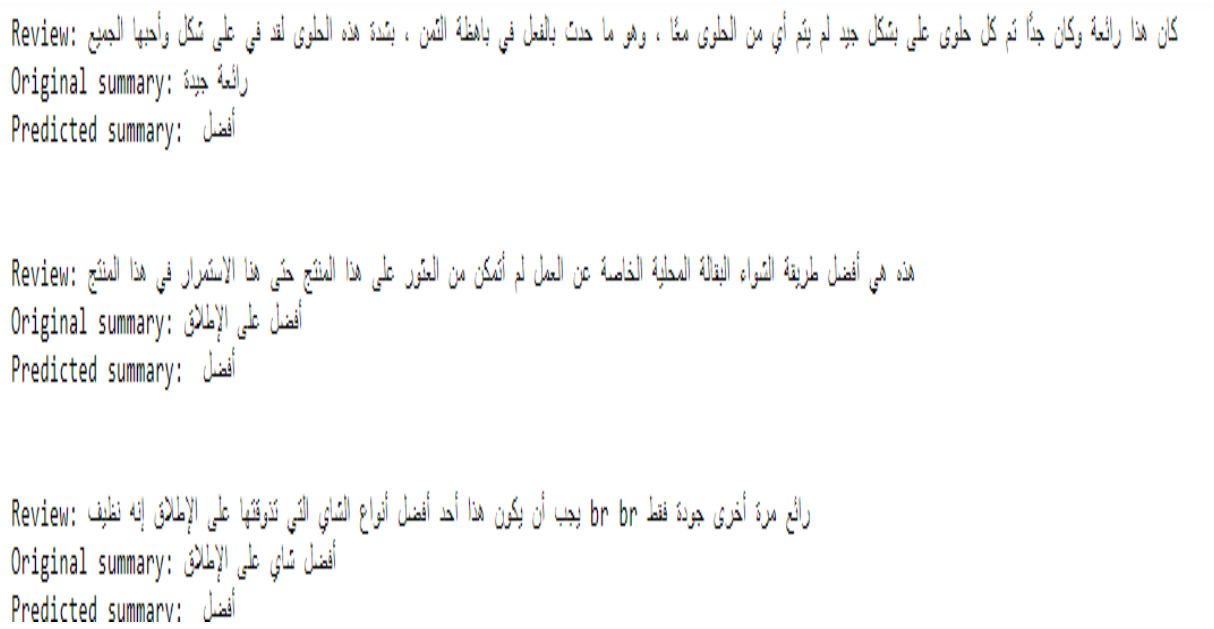


Figure 4.7: Results from ATS Arabic-Arabic.

4.4.8 Results from ATS Arabic-English

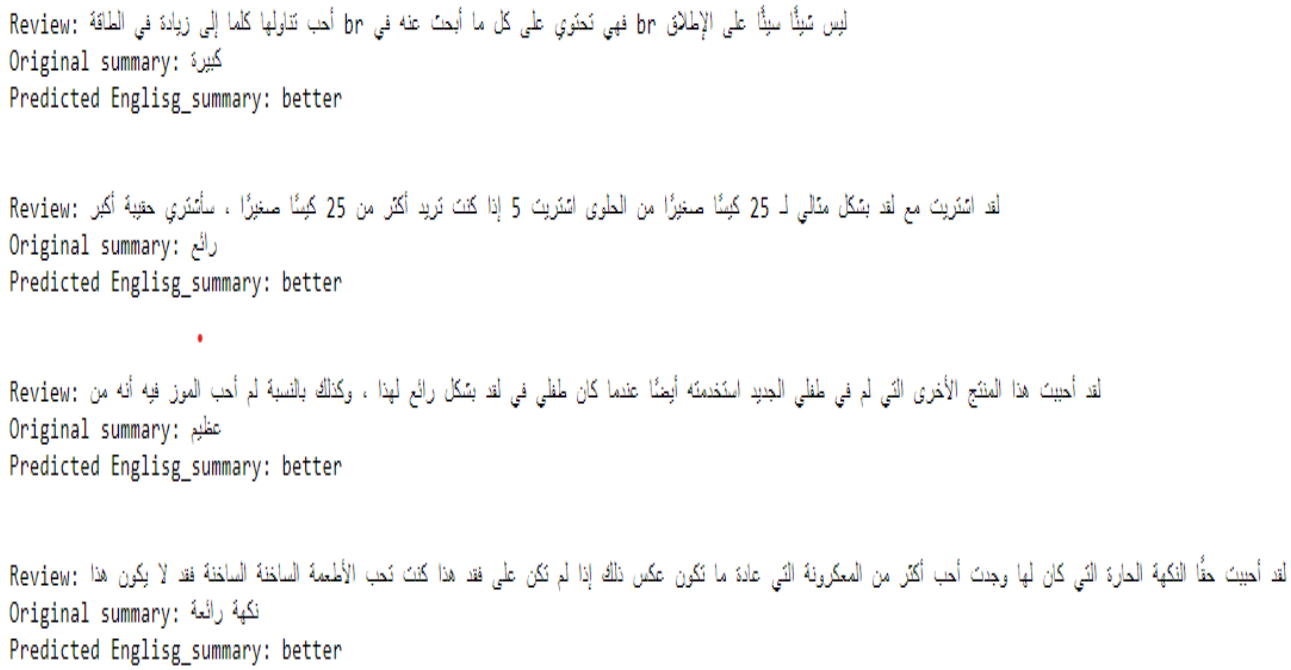


Figure 4.8: Results from ATS Arabic-English.

4.4.9 Results from ATS Arabic-French

Review: لقد اشتريت مع لقد بشكل مثالي لـ 25 كيسًا صغيرًا من الحلوى اشتريت 5 إذا كنت تريد أكثر من 25 كيسًا صغيرًا ، سأشتري حقيبة أكبر
Original summary: رائع
Predicted French_summary: Merveilleux

Review: لقد أحببت هذا المنتج الأخرى التي لم في طفلي الجديد استخدمته أيضًا عندما كان طفلي في لقد بشكل رائع لهذا ، وكذلك بالنسبة لم أحب الموز فيه أنه من
Original summary: عظيم
Predicted French_summary: Merveilleux

Review: لقد أحببت حقًا النكهة الحارة التي كان لها وجدت أحب أكثر من المعكرونة التي عادة ما تكون عكس ذلك إذا لم تكن على قد هذا كنت تحب الأطعمة الساخنة الساخنة فقد لا يكون هذا
Original summary: نكهة رائعة
Predicted French_summary: Merveilleux

Figure 4.9: Results from ATS Arabic-French.

4.4.10 Results from ATS Train Arabic-Test French

Review: العمل ، مع لأي إذا كنت تبحث عن هذه الصغيرة كما لو كنت من قبل العبوة على الأخرى ، فهذا هو أفضل لك على ، فإن سعر للغاية
Original summary: super
Predicted summary: bien

Review: ولكن المذاق جودة التجارية تقريبًا اعتقدت أن هذه الأشياء بسرعة ، لكن الليمون التي أحبها ما لم هذا منتج جيد فق ط أتمنى لو كان لا يزال قد في كما كان
Original summary: doux délicieux
Predicted summary: super

Review: الدجاج وقتًا أطول في الطهي الدجاج فوق الخضار إلى لقد مع فقط أنا لا أحب الذرة الصغيرة بعض رقائق الفلفل الأحمر فوق الأرز رائع على في أقل من 30 دقيقة
Original summary: poulet aime pas
Predicted summary: mauvais

Review: لقد أحببت هذا المنتج الأخرى التي لم في طفلي الجديد استخدمته أيضًا عندما كان طفلي في لقد بشكل رائع لهذا ، وكذ لك بالنسبة لم أحب الموز فيه أنه من المستحيل
Original summary: bon produit
Predicted summary: super

Figure 4.10: Results from ATS Train Arabic-Test French.

4.4.11 Results from ATS Train Arabic-Test English

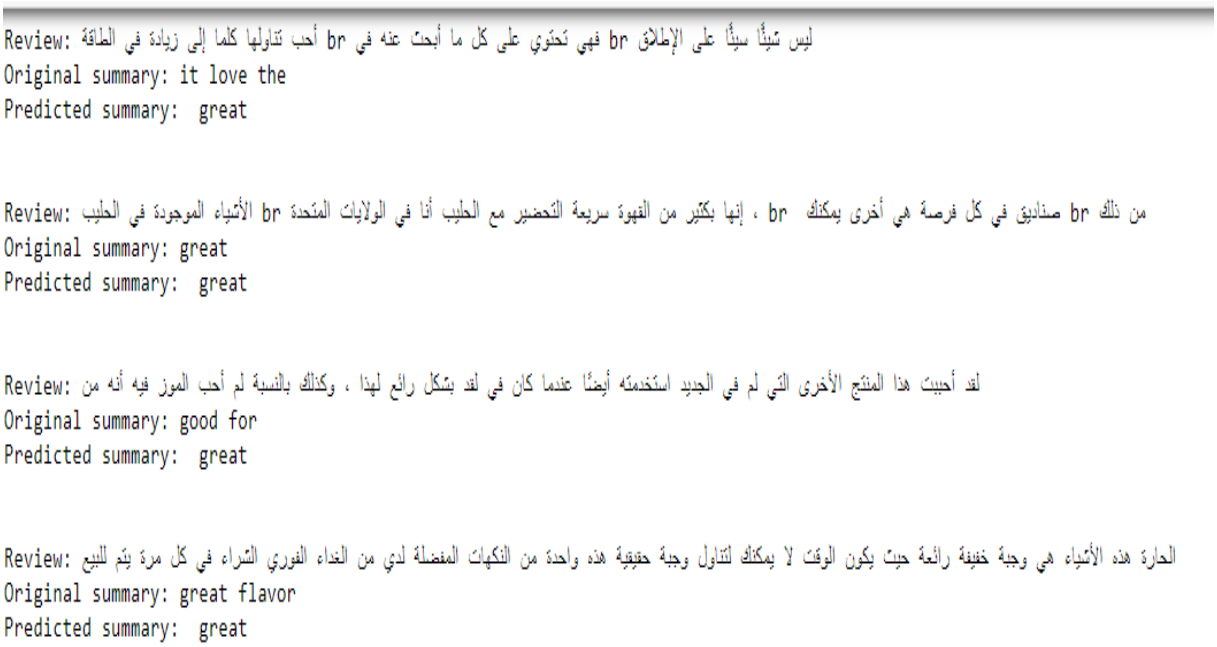


Figure 4.11: Results from ATS Train Arabic-Test English.

4.4.12 Results from ATS Train French- Test English

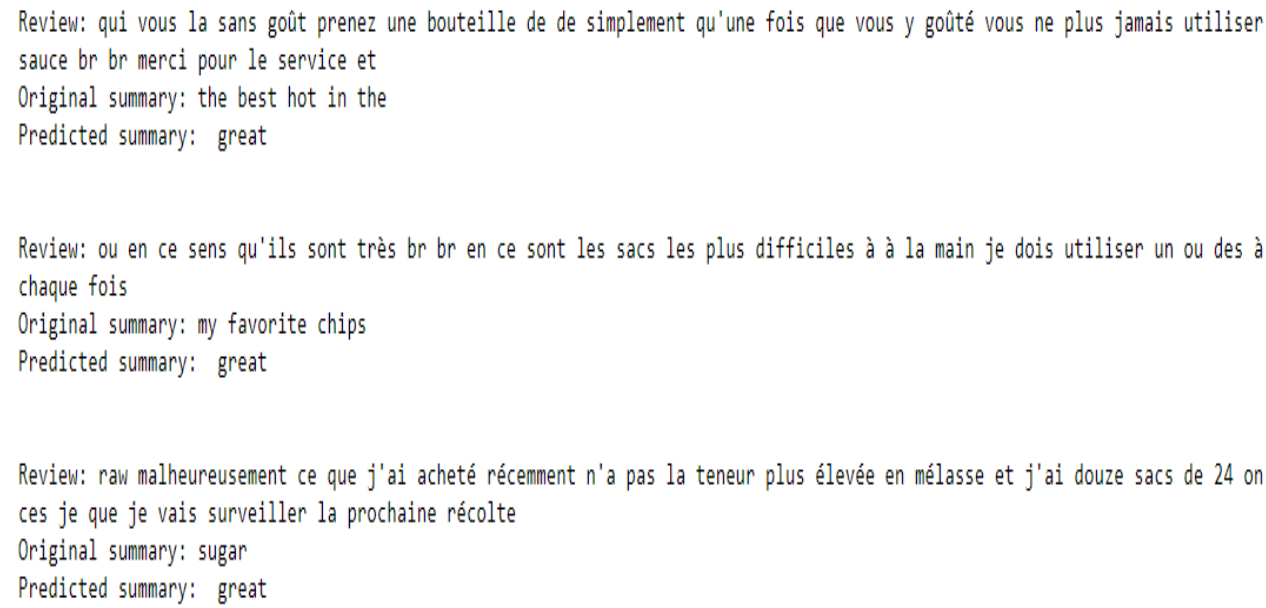


Figure 4.12: Results from ATS Train French-Test English.

4.4.13 Results from ATS Train French- Test Arabic

```
Review: c'est beaucoup plus et plus savoureux que le café instantané au lait je suis et aux br et des de à chaque est un autr
e que vous pouvez br le
Original summary: رائع
Predicted summary:  رائع


Review: de problème cependant les bananes en ont à peu près quelques unes bien que j'aime ce produit je pense que la prochain
e fois celui qui a les sacs en et
Original summary: عظيم
Predicted summary:  رائع


Review: temps est et parfois vous ne pouvez pas vous pour un vrai repas c'est l'une de mes saveurs préférées et je acheter à
chaque fois qu'il sera mis en vente
Original summary: نكهة رائعة
Predicted summary:  رائع
```

Figure 4.13: Results from ATS Train French- Test Arabic.

4.4.14 Results from ATS Train English-Test French

```
Review: i have been using the product for several months now the dogs seem to like it and having it delivered is really nice
what more can i say
Original summary: bon produit
Predicted summary:  bon


Review: this will sure does throw the kcup in and this is the best hot choc ever i choke down my vitamins with it too worth i
t and so cheap here
Original summary: plus jamais une autre marque
Predicted summary:  super


Review: store it in a bowl or else i'm up water from my fridge shelf on the plus side the bottom helps to remind me to change
the water every day
Original summary: produit s'il était plus facile à ouvrir
Predicted summary:  super
```

Figure 4.14: Results from ATS Train English-Test French**.**

4.4.15 Results from ATS Train English-Test Arabic

Review: i can't stop eating them it's crunchy and seasoned at the right amount of flavor and salt just perfect believe me som
e chips are so salty i can't finish a small bag but this one i just keep on eating
Original summary: أحب هذه الرقائق
Predicted summary: أفضل خفيفة على الإطلاق

Review: cake like this at our local asian supermarkets so i ordered some from here these lovely made our party much more fun
my friends and i think it is a good idea to order them again for our mid party
Original summary: لذيذ جدا لكن احذر
Predicted summary: طعام رائع

Review: puppy 4 threw up castor and and natural balance and had loose so i tried this and he did great even the red from his
eyes have almost cleared up after a month along with eye drops which is great
Original summary: هذا هو أفضل طعام للكلاب
Predicted summary: طعام رائع

Figure 4.15: Results from ATS Train English-Test Arabic.

4.5 The results of the Bleu and Rouge scores

4.5.1 The results of the Bleu score

Input	Output	Bleu score
English	English	0.50577
English	Arabic	0.50495
English	French	0.50550
French	French	0.47114
French	Arabic	0.38892
French	English	0.40534
Arabic	Arabic	0.40667
Arabic	English	0.40836
Arabic	French	0.42914

Table 4.1: Results of the Bleu score.

The BLEU score in ATS (English-English) is higher than in ATS (Arabic-Arabic) and ATS (French-French), because the dataset (*Reviews*) is bigger compared with other datasets. The generalization capability of a deep learning model is enhanced with an increase in the training dataset size.

Input training	Input testing	Output	Bleu score
Arabic	English	English	0.45883
Arabic	French	French	0.46079
French	English	English	0.47237
French	Arabic	Arabic	0.43042
English	Arabic	Arabic	0.44725
English	French	French	0.50351

Table 4.2: Results of the Bleu score (different input).

4.5.2 Sentence BLEU score for validation dataset

We calculated the blue score for validation sentences as shown in the figures below:

- **Sentence BLEU score for validation dataset (English-English):**

```
ref : the best rice drink , prediction:  love this stuff, score: 0.355945839284367

ref : illy dark roast espresso pods , prediction:  great espresso, score: 0.3789614601174753

ref : not the same , prediction:  not so good, score: 0.3510775154699537

ref : arrived on time and in advertised condition , prediction:  great product, score: 0.38219883874210236

ref : omg these are the bomb , prediction:  not the best, score: 0.3739151740699424

ref : convenient , prediction:  cats love it, score: 0.3739151740699424

ref : delicous , prediction:  delicious, score: 0.41763216744945864

ref : so so but not something need , prediction:  not good, score: 0.37713256337079715

ref : these were beyond heavenly , prediction:  the best, score: 0.37713256337079715

ref : larabars are staple , prediction:  great snack, score: 0.36440899897551626
```

Figure 4.16: Results of the Bleu score (English-English).

- **Sentence BLEU score for validation dataset (English-French):**

ref : scrumptious , prediction: délicieux, score: 0.32462292658309544

ref : love this stuff , prediction: super, score: 0.33395396479149164

ref : taste great , prediction: bon produit, score: 0.3237952642889745

ref : works and tastes good , prediction: bon produit, score: 0.3677577795437005

ref : delicious mac cheese meal , prediction: super macaroni au fromage, score: 0.369282183087816

ref : new staple for my pantry , prediction: chaud chaud chaud, score: 0.31463273706111694

ref : this milk is really good , prediction: super pour le prix, score: 0.3626415455488691

ref : smoked salmon , prediction: bon produit, score: 0.3387188167521856

ref : chocolates , prediction: délicieux, score: 0.32462292658309544

Figure 4.17: Results of the Bleu score (English-French).

- **Sentence BLEU score for validation dataset (English-Arabic):**

ref : the best rice drink , prediction: أحب هذه الأشياء, score: 0.3009822437761237

ref : illy dark roast espresso pods , prediction: نكهة رائعة, score: 0.29033122409548967

ref : not the same , prediction: أحب هذه الأشياء, score: 0.3009822437761237

ref : arrived on time and in advertised condition , prediction: منتج جيد, score: 0.2843285192944476

ref : omg these are the bomb , prediction: جبنة ماك كبيرة, score: 0.29919647573412606

ref : convenient , prediction: صلصة حارة كبيرة, score: 0.3009822437761237

ref : delicous , prediction: لذيذ وصحي, score: 0.2875080666456396

Figure 4.18: Results of the Bleu score (English-Arabic).

- **Sentence BLEU score for validation dataset (French-English):**

```
ref : grand , prediction: Well, score: 0
ref : très bien , prediction: Well, score: 0.2650527868304097
ref : idéal pour la , prediction: Well, score: 0.2650527868304097
ref : saveur , prediction: Well, score: 0.2650527868304097
ref : mes chats ne sont pas de la nourriture , prediction: Well, score: 0.3037563786393773
ref : super , prediction: Well, score: 0.2650527868304097
ref : doux délicieux et , prediction: Well, score: 0.3037563786393773
ref : pas un thé , prediction: Well, score: 0
ref : bon pour la , prediction: Well, score: 0.2650527868304097
```

Figure 4.19: Results of the Bleu score (French-English).

- **Sentence BLEU score for validation dataset (Arabic-Arabic):**

```
ref : كبيرة , prediction:  أفضل, score: 0.2713915953910661
ref : رائع , prediction:  أفضل, score: 0.2713915953910661
ref : عظيم , prediction:  أفضل, score: 0.2713915953910661
ref : نكهة رائعة , prediction:  أفضل, score: 0.2713915953910661
ref : من , prediction:  أفضل, score: 0.2713915953910661
ref : رائع , prediction:  أفضل, score: 0.2713915953910661
ref : ، ، لذيذ , prediction:  أفضل, score: 0.30325506570703464
ref : ليس شاي , prediction:  أفضل, score: 0.30325506570703464
```

Figure 4.20: Results of the Bleu score (Arabic-Arabic).

- **Sentence BLEU score for validation dataset (Train Arabic-Test English):**

```
ref : it love the , prediction:  great, score: 0.3297680972270839
ref : great , prediction:  great, score: 0.4047136050061484
ref : good for , prediction:  great, score: 0.3297680972270839
ref : great flavor , prediction:  great, score: 0.4047136050061484
ref : my cats are not of the food , prediction:  great, score: 0.3801955718365812
ref : great , prediction:  great, score: 0.4047136050061484
ref : delicious and sweet , prediction:  great, score: 0.3552507849445419
ref : not a tea , prediction:  great, score: 0.3552507849445419
ref : for cats , prediction:  great, score: 0.3552507849445419
ref : in , prediction:  great, score: 0.2764615358595646
```

Figure 4.21: Results of the bleu score (Train Arabic-Test English).

- **Sentence BLEU score for validation dataset (Train English-Test Arabic):**

```
ref : فطائر الحبوب , prediction:  أفضل خفيفة على الإطلاق, score: 0.3424497522308629
ref : عرق السوس , prediction:  لذيذ, score: 0.30325506570703464
ref : نيومان , prediction:  أفضل خفيفة على الإطلاق, score: 0.32732017521541856
ref : منعش مزيج لذيذ منزوع الكافيين , prediction:  أفضل خفيفة على الإطلاق, score: 0.34987552277302525
ref : هذه الرقائق رائعة لم الأفضل ولكن , prediction:  أفضل مزيج على الإطلاق, score: 0.37309164765294484
ref : بذور أو الخبز أو , prediction:  أفضل خفيفة على الإطلاق, score: 0.3424497522308629
```

Figure 4.22: Results of the bleu score (Train English-Test Arabic).

- **Sentence BLEU score for validation dataset (Train English-Test French):**

```
ref : bon produit , prediction:  bon, score: 0.37727692088527237
ref : plus jamais une autre marque , prediction:  super, score: 0.4047136050061484
ref : produit s'il était plus facile à ouvrir , prediction:  super, score: 0.4047136050061484
ref : à trouver , prediction:  bon, score: 0.3037563786393773
ref : soja mais pas bon comme cadeau unique , prediction:  bon, score: 0.37727692088527237
ref : j'adore ces chips , prediction:  bon, score: 0.3037563786393773
ref : pas cher , prediction:  bon, score: 0.2650527868304097
ref : très savoureux mais attention , prediction:  super, score: 0.3801955718365812
```

Figure 4.23: Results of the bleu score (train English-Test French).

- **Sentence BLEU score for validation dataset (Train Arabic-Test French):**

```
ref : bon , prediction: bon, score: 0.30601761630731034
ref : excellent , prediction:  bon, score: 0.30601761630731034
ref : frais parfait , prediction:  bien, score: 0.25647294964174183
ref : meilleure , prediction:  bon, score: 0.3537632202021088
```

Figure 4. 24: Results of the bleu score (Arabic-Test French).

- **Sentence BLEU score for validation dataset (Arabic-French):**

```
ref : كبيرة , prediction:  رائع, score: 0.30325506570703464
ref : رائع , prediction:  رائع, score: 0.3930776032102183
ref : عظيم , prediction:  رائع, score: 0.30325506570703464
ref : نكهة رائعة , prediction:  رائع, score: 0.3930776032102183
ref : من , prediction:  رائع, score: 0.2713915953910661
ref : رائع , prediction:  رائع, score: 0.3930776032102183
```

Figure 4.25: Sentence BLEU score for validation dataset (Arabic-French).

- **Sentence BLEU score for validation dataset (French-French):**

```
ref : très bien , prediction:  bon, score: 0.30601761630731034
ref : saveur , prediction:  super, score: 0.30601761630731034
ref : super , prediction:  bien, score: 0.30601761630731034
ref : doux délicieux , prediction:  super, score: 0.3537632202021088
```

Figure 4.26: Sentence BLEU score for validation dataset (French-French).

- **Sentence BLEU score for validation dataset (Train French-Test Arabic)**

```
ref : كبيرة , prediction:  رائع, score: 0.30325506570703464
ref : رائع , prediction:  رائع, score: 0.3930776032102183
ref : عظيم , prediction:  رائع, score: 0.30325506570703464
ref : نكهة رائعة , prediction:  رائع, score: 0.3930776032102183
ref : من , prediction:  رائع, score: 0.2713915953910661
ref : رائع , prediction:  رائع, score: 0.3930776032102183
ref : ، ، حلو ، لذيذ , prediction:  أفضل, score: 0.30325506570703464
```

Figure 4.27: Results of the bleu score (train Arabic-Test Arabic).

- **Sentence BLEU score for validation dataset (Train French-Test English)**

```
ref : great , prediction:  great, score: 0.4047136050061484
ref : great , prediction:  great, score: 0.4047136050061484
ref : great for , prediction:  great, score: 0.4047136050061484
ref : great flavor , prediction:  great, score: 0.4047136050061484
ref : my cats are not of the food , prediction:  great, score: 0.3801955718365812
ref : great , prediction:  great, score: 0.4047136050061484
ref : delicious and sweet , prediction:  great, score: 0.3552507849445419
```

Figure 4.28: Sentence BLEU score for validation dataset (Train French-Test English).

- **Sentence BLEU score for validation dataset (Arabic -English)**

```
ref : كبيرة , prediction:  أفضل, score: 0.2713915953910661
ref : رائع , prediction:  أفضل, score: 0.2713915953910661
ref : عظيم , prediction:  أفضل, score: 0.2713915953910661
ref : نكهة رائعة , prediction:  أفضل, score: 0.2713915953910661
ref : من , prediction:  أفضل, score: 0.2713915953910661
ref : رائع , prediction:  أفضل, score: 0.2713915953910661
```

Figure 4.29: Sentence BLEU score for validation dataset (Arabic-English).

- **Sentence BLEU score for validation dataset (French-Arabic)**

```
ref : très bien , prediction: جيد , score: 0.2764615358595646
ref : saveur , prediction: جيد, score: 0.2764615358595646
ref : super , prediction: ممتاز, score: 0.2764615358595646
ref : doux délicieux , prediction: جيد, score: 0.2764615358595646
```

Figure 4. 30: Sentence BLEU score for validation dataset (French-Arabic).

4.5.2 The results of the Rouge score

Rouge	Precision	Recall	F-measure
Rouge-1	19.29	14.67	15.63
Rouge-2	4.24	3.27	3.47
Rouge-3	0.60	0.50	0.52
Rouge-4	0 .06	0.04	0.05
Rouge -L	20.71	16.36	17.45

Table 4.3: Results of the Rouge score.

4.6 Answers to Research Questions

The research questions presented in Chapter 1 have been investigated and researched throughout this thesis. Below we have summarized the answers to these questions based on the findings confirmed by this study.

4.6.1 Answer 1

- The Encoder-Decoder architecture is mainly used to solve the sequence-to-sequence (Seq2Seq) problems where the input and output sequences are of different lengths.

4.6.2 Answer 2

- The solution encoder to memorize long sequences into a fixed-length vector is LSTM (Long Short-Term Memory) because it is capable of capturing long-term dependencies by overcoming the problem of vanishing gradient.

4.6.3 Answer 3

- The basic idea behind the attention mechanism**,** So, instead of looking at all the words in the source sequence, we can increase the importance of specific parts of the source sequence that result in the target sequence.

4.6.4 Answer 4

- We minimized the loss and saved time in memory as well as computation by using *sparse_categorical_crossentropy* as the loss function, since it converts the integer sequence to a one-hot vector on the fly. This overcomes any memory issues.

4.7 Conclusion

We presented the results of our model ATS using multiple datasets and different inputs. These results reveal that if the dataset is higher, the domain knowledge will be higher, so the predicted summary will be higher quality. We will get good results. As for the sentence, the longer the sentence, the less bleu score, and vice versa. Finally, we presented the answers to the problems we mentioned before.

Conclusion

General Conclusion and perspective

Today, with all the digitalization and with the internet being the endless source of text data, we are kind of flooded with textual information, and it's really hard to find relevant information in this amount of text. So, automatic text summation has become a necessary need.

In this work, we explored summarization. To do so, we had to perform summarization in an abstractive way, i.e., by creating completely new sentences.

We tried to tackle problems which appear when meeting unknown words or misspelled words. To do so, we implemented a network to perform character level summarization. However, thinking summarization at a character level made the task of understanding the meaning of a sentence even harder.

The results of our approach revealed that BLEU scored 0.50, which is very good. We used our approach to cross-language abstractive text summarization and the result was very good. The BLEU score is 50

As a perspective, to enhance the performance of the proposed models of abstractive text summarization using encoder-decoder, RNN, and attention mechanisms, we recommend increasing the training dataset size and building the model. The generalization capability of a deep learning model is enhanced with an increase in the training dataset size and Try implementing Bi-Directional LSTM, which is capable of capturing the context from both directions and results in a better context vector. Use the beam search strategy for decoding the test sequence instead of using the greedy approach (argmax). Implement pointer-generator networks and coverage mechanisms.

REFERENCES

[1] Dzmitry Bahdanau,KyungHyun Cho , Yoshua Bengio , NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE https://arxiv.org/pdf/1409.0473.pdf

[2] Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need

[3] Sumit Chopra , Michael Auli, Alexander M. Rush Abstractive Sentence Summarization with Attentive Recurrent Neural Networks https://aclanthology.org/N16-1012.pdf

[4] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos , Bing Xiang, Çaglar G ˘ ulçehre Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond https://arxiv.org/pdf/1602.06023.pdf

[5] Abigail See, Peter J. Liu, Christopher D. Manning, Get To The Point: Summarization with Pointer-Generator Networks https://arxiv.org/pdf/1704.04368.pdf

[6] Shengli Song, Haitao Huang ,Tongxiao Ruan Abstractive text summarization using LSTM-CNN based deep learning https://link.springer.com/content/pdf/10.1007/s11042-018-5749-3.pdf

[7] Yiming Cui , Zhipeng Chen , Si Wei , Shijin Wang , Ting Liu‡ and Guoping Hu Attention-over-Attention Neural Networks for Reading Comprehension https://arxiv.org/pdf/1607.04423.pdf

[8] Panagiotis Kouris , Georgios Alexandridis , Andreas Stafylopatis Abstractive text summarization based on deep learning and semantic content generalization https://aclanthology.org/P19-1501.pdf

[9] Dipanjan Das and André FT Martins. A survey on automatic text summarization, 2007, 2007.

[10] Juan-Manuel Torres-Moreno. Automatic text summarization. John Wiley & Sons, 2014.

[11] Extractive Summarization using Deep Learning https://arxiv.org/pdf/1708.04439.pdf.

[12] Mohamed Hedi Maaloul. Approche hybride pour le résumé automatique de textes. Application à la langue arabe. PhD thesis, 2012

[13] NazariandMA Mahdavi.A survey on automatic text summarization.Journal of AI andDataMining,7(1):121–135,2019.

[14] Rafael Ferreira, Luciano de Souza Cabral, Rafael DueireLins,GabrielPereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J Simske, and Luciano Favaro. Assessing sentence scoring techniques for extractive text summarization. Ex- pert systems with applications, 40(14):5755–5764, 2013.

[15] WafaaS El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. Expert Systems with Applications, page 113679,2020.

[16] Florian Boudin, Juan-Manuel Torres-Moreno, and Marc El-Béze. Improving update summarization byrevisitingthemmr criterion.arXiv preprint arXiv:1004.3371,2010.

[17] Abdelkrim Aries,WalidKhaledHidouci,et al.Automatic Text Summarization:What hasbeendone andwhathastobedone.arXivpreprint arXiv:1904.00688,2019.

[18] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. Artificial Intelligence Review, 47(1):1–66, 2017.

[19] Adhika Pramita Widyassari , Supriadi Rustad , Guruh Fajar Shidik , Edi Noersasongko. Review of automatic text summarization techniques & methods.

[20] Josef Steinberger and Karel Ježek. Evaluation measures for text summarization. *Com- putingandInformatics*,28(2):251–275,2012.

[21] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," pp. 311–318, 2002.

[22] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Text Summarization Branches Out, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[23] K. Kaikhah, "Automatic text summarization with neural networks," pp. 40 – 44 Vol.1, 07 2004.

[24] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," arXiv e-prints, p. arXiv:1509.00685, Sep 2015.

[25] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," pp. 93–98, 01 2016.

[26] R. Nallapati, B. Zhou, C. Nogueira dos santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," arXiv e-prints, p. arXiv:1602.06023, Feb 2016.

[27] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," 2017.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv e-prints, p. arXiv:1301.3781, Jan 2013.

[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," arXiv e-prints, p. arXiv:1310.4546, Oct 2013.

[30] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to Generate Reviews and Discovering Sentiment," arXiv e-prints, p. arXiv:1704.01444, Apr 2017.

[31] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[32] https://aclanthology.org/2020.acl-main.121.pdf.

[33] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

[34] Goodfellow, I, Bengio, & Courville, A. (2016). *Deep Learning*. MIT Press. https://books.google.dz/books?id=Np9SDQAAQBAJ

[35] Dessí, D., Dragoni, M., Fenu, G., Marras, M., & Reforgiato Recupero, D. (2020). Deep Learning Adaptation with Word Embeddings for Sentiment Analysis on Online Course Reviews. In Deep Learning-Based Approaches for Sentiment Analysis (pp. 57–83). Springer. https://doi.org/10.1007/978-981-15-1216-2_3

[36] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

[37] Nankani, H., Dutta, H., Shrivastava, H., Rama Krishna, P. V. N. S., Mahata, D., & Shah, R. R. (2020). Multilingual Sentiment Analysis. In Deep Learning-Based Approaches for Sentiment Analysis (pp. 193–236). Springer. https://doi.org/10.1007/978-981-15-1216-2_8

[38] Bengio, Yoshua, Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, *5*(2), 157–166.

[39] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[40] TORRES.AI, J. (2018). Learning process of a neural network How Do Artificial Neural Networks Learn? Sep 21, 2018. https://towardsdatascience.com/how-doartificial-neural-networks-learn-773e46399fc7

[41] Gómez, R. (2018). Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names. May 23, 2018. https://gombru.github.io/2018/05/23/cross_entropy_loss/

[42] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Doll'ar, P. (2017). Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision, 2980–2988.

[43] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv Preprint ArXiv1409.0473. http://arxiv.org/abs/1409.0473

[44] Ahmet, A., & Abdullah, T. (2020). Recent Trends and Advances in Deep Learning-Based Sentiment Analysis. In Deep Learning-Based Approaches for Sentiment Analysis (pp. 33–56). Springer. https://doi.org/10.1007/978-981-15-1216-2_2