
SAARLAND UNIVERSITY

Faculty of Mathematics and Computer Science
Department of Computer Science
MASTER THESIS



A Modular Model for Cross-lingual Image Captioning

submitted by
Tanvi Ajay Gunjal
Saarbrücken
March 2024

Advisor:

Mareike Hartmann, Ph.D
Saarland University
Saarbrücken, Germany

Reviewers

Prof. Dr. Daniel Sonntag
German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus
Saarbrücken, Germany

Prof. Dr. Antonio Krüger
German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus
Saarbrücken, Germany

Saarland University
Faculty MI – Mathematics and Computer Science
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Declarations

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,

(Datum/Date)

(Unterschrift/Signature)

Acknowledgements

I would like to sincerely express my gratitude to my advisor, Mareike Hartmann, for her unwavering guidance, insightful feedback, and unflinching assistance during the entire duration of this research. Her profound knowledge in the domain and guidance were crucial in determining the course and accomplishment of the work. I would like to extend my gratitude to my supervisors, Prof. Dr. Antonio Krüger and Prof. Dr.-Ing. Daniel Sonntag, for their encouragement, valuable suggestions, and constructive criticism that significantly contributed to the refinement of the study.

A sincere thank you to the examination office for their cooperation. Special thanks to Prof. Dr. Jens Dittrich and Ms. Silke Lorang for administrative support and assistance in navigating the procedural aspects of this research.

I would like to express my gratitude to my friend Akshay Joshi, whose encouragement and shared passion for machine learning made the research journey more enjoyable and rewarding. I am always grateful to my parents and my family for their active encouragement and motivation. I am incredibly grateful to them for their support, especially during challenging days. I also want to thank my friends for their support and kindness.

Abstract

Image captioning is a challenging task in the domain of computer vision and natural language processing, with the goal of developing an algorithm that can automatically generate informative and contextually relevant captions for images. The task involves the fusion of multi-modal elements, i.e., visual perception and language understanding, requiring the model to extract relevant features from the image and translate them into textual descriptions. Recent works in image captioning has shown that the models use an encoder-decoder architecture [78], [84], [42], [45] and most of the work has been extensively studied in English, primarily because of the availability of the data. Nevertheless, the advantages of this advanced technology are not useful to a significant portion of those who do not speak English. The same captioning model can be applicable for non-English languages as long as there is sufficient training data available [61], [28]. However, the lack of training data for non-English target languages is a notable challenge. In such situations, it is necessary to make use of alternate resources, such as unpaired data or manually translating captions in the target language. To overcome this limitation, the study employs a strategic approach called cross-lingual transfer learning. This approach leverages information gathered from a resource-rich source language to improve performance in a target language that has limited image-caption-paired data. Therefore, the thesis specifically investigates cross-lingual transfer learning to generate captions in a non-English target language, especially when there is a limited availability of data. Instead of relying solely on target language resources, the study integrates additional sources such as image pairs with English captions and parallel sentences (sentences in two languages that are translations of each other). To achieve this, the research extends a modular approach previously proposed for the machine translation task by Lyu et al. [57], to address the unique challenges posed by the image captioning task. By employing this architecture, our aim is to exploit the Cross-language effect, which occurs when multiple languages are merged into a common module. It has been observed that low-resource languages can gain significant advantages from high-resource image-caption-pair data. This strategic approach aims to improve the efficiency and flexibility of the model, especially in situations where there is a shortage of language resources. This extension represents a novel contribution to the field, providing a nuanced and adaptable framework for multilingual image captioning under data constraints.

Keywords: Pattern Recognition, Machine Learning, Deep learning, Computer Vision, Natural Language Processing, Multi-linguality.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research problem	3
1.3	Research objectives	3
1.4	Research methodologies	4
1.5	Scope and limitations	4
1.6	Contribution	5
1.7	Outline of the thesis	5
2	Concept	7
2.1	Preliminaries	7
2.1.1	Artificial Intelligence (AI) and Machine Learning (ML)	7
2.1.2	Deep Learning (DL)	9
2.1.3	Convolution Neural Networks (CNN)	11
2.1.4	Recurrent Neural Networks (RNN)	15
2.1.5	Transformers	18
3	Related Work	23
3.1	Image Captioning (IC)	23
3.1.1	Neural Image Captioning	23
3.1.2	Cross-Lingual strategies in Image Captioning	26
3.2	Few-shot Image Captioning	29
3.3	Neural Machine Translation (NMT)	30
3.4	Summary	32
4	Implementation	33
4.1	Methodology	33
4.1.1	Encoder Architecture	33
4.1.2	CNN-LSTM Architecture	34
4.1.3	CNN-Transformer Architecture	36
4.2	Data Pre-processing	39
4.3	Approaches	41

4.3.1	Single model	41
4.3.2	1-1 model	42
4.3.3	M2 architecture	46
4.4	Training details	48
4.4.1	Loss Function	48
4.4.2	Beam Search	48
4.4.3	Early stopping	48
4.4.4	Training details	50
4.4.5	Hyper-parameter tuning and Computational Setup	51
4.4.6	Training Durations	51
5	User Study	53
5.1	Dataset	53
5.1.1	MS-COCO-2014 Dataset	53
5.1.2	MS-COCO-it Dataset	55
5.1.3	MS-COCO-ES Dataset	56
5.1.4	Comparison between English, Italian and Spanish dataset	56
5.2	Evaluation Metrics	57
5.2.1	n-grams	60
5.2.2	BLEU (Bilingual Evaluation Understudy)	60
5.2.3	ROUGE (Recall Oriented Understudy for Gisting Evaluation)	62
5.2.4	CIDEr (Consensus-based Image Description Evaluation)	63
5.3	Results	64
5.3.1	Research Question: 01	64
5.3.2	Research Question: 02	66
5.3.3	Research Question: 03	67
5.3.4	Research Question: 04	70
5.3.5	Research Question: 05	73
5.3.6	Research Question: 06	76
6	Conclusion	83
6.1	Conclusion	83
6.2	Limitations	84
6.3	Future works	84
	Bibliography	86
A	Appendix	95

List of Figures

1.1	An example for image captioning task (Source: [7])	2
2.1	Relationship between Artificial Intelligence, Machine Learning, Deep Learning and Natural Language Processing (Source: [60])	8
2.2	Schematic representation detailing the structure of a single perceptron, showcasing its neuron and connections in the neural network framework (Source:)	10
2.3	A CNN is made up of two primary parts. The process of feature learning and the subsequent classification component. In the feature learning phase, a series of convolutional layers are sequentially arranged to acquire knowledge of features, beginning with fundamental ones such as edges and progressing towards complex ones. The classification layer is comprised of a series of fully connected (FC) layers that allocate the input to specific classes (Source: [1]).	12
2.4	Illustration of feature extraction using convolutional filters or kernels (Source: [3])	13
2.5	The concept of residual connection is employed in the ResNet architecture, as described in the work by ABC. The fundamental concept of residual learning pertains to the optimization of a stack of layers by learning a residual mapping $F(x)$ instead of the original mapping $H(x)$. This is achieved by expressing $H(x)$ as the sum of $F(x)$ and x (Source: [11]) . . .	14
2.6	A representation of an RNN is shown on the left side, and an RNN that has been unfolded (or unrolled) into a full network is shown on the right side. The term "unrolling" refers to the process of representing the network by explicitly writing out its structure for the entire series. For instance, in the case when the sequence of interest is a sentence consisting of three words, the neural network would be expanded into a network with three layers, with each layer corresponding to a specific word. (Source: [2])	15
2.7	An LSTM Block contains four interacting layers (cell state, an input gate, a forget gate, an output gate) (Source:[14])	17
2.8	Multi-Head Attention with h attention heads, all running in parallel. (Source: [80])	19
2.9	The Transformer model architecture. The encoder consists of N blocks on the left, while the decoder consists of N blocks on the right. (Source:[13]) .	21
3.1	Approach Overview: In step (2), lower convolutional layers capture image features. Step (3), a feature is sampled and input to an LSTM to generate the corresponding word. Step 3 is iteratively repeated K times to produce a K -words caption. (Source: [84])	24
3.2	A unified model for multilingual captioning, employing artificial tokens to facilitate language switching (Source: [75])	28

4.1	CNN Encoder - ResNet-101; Processes an image with 3 color channels, outputting feature maps through convolutional blocks	34
4.2	CNN- LSTM approach: Image features are captured at lower convolutional layers, sampled, and fed to LSTM for generating corresponding words. This process is repeated K times to produce a K-words caption	35
4.3	A high-level overview of an image captioning system which incorporates a ResNet-101 model as image Encoder and Marian NMT model as auto-regressive text Decoder	37
4.4	An expanded illustration of the modifications made to the Marian NMT auto-regressive text decoder. The section inside the dashed lines illustrate the addition of 1D convolution layers to extract information from the feature maps passed by the encoder	38
4.5	Left-most section: convert the RGB image into grey scale. Right-most section: Converts the raw captions into encoding. Mid section: pre-processed images and captions are serialized and stored in binary file format such as HDF5	40
4.6	Overview of three distinct multilingual image captioning models designed for the languages English (En), Spanish (Es), and Italian(it). Left-most section: consists of a collection of individual models designed for 3 different directions. Mid section: a 1-1 model that shares all the parameters of the model . Reight-most section: the M2 approach primarily shares language-specific modules.	42
4.7	Data Pre-processing: inject language token at the beginning of the captions to instruct the model to generate caption in the target language	43
4.8	Data pre-processing: overlay language token on the image in three different positions with two different colors, to guide the model generate caption in target language	45
4.9	An example explaining cross-modal grounding. The left-most section: model generates a accurate captions extracting information from multiple modalities. The right-most section: the model generates inaccurate caption, which suggests that the model lacks the ability of extract information from both modalities	46
4.10	Left-most section: NMT model pre-tainting, an encoder-decoder architecture to train a German-English translation model and using the same German encoder to train a German-Spanish translation model. Mid section: Train a image captioning system, with image encoder and English NMT decoder. Left-most section: use the pre-trained image encoder and pre-trained NMT Spanish decoder to perform few-shot learning	47
4.11	Illustration of Beam Search: A graphical depiction illustrating the consecutive stages of beam search. It demonstrates the multiple potential paths to generate more accurate and contextually relevant sequences	49
5.1	Distribution of Data Classes for the MS-COCO dataset: Illustrating the relative frequencies of different classes within the dataset	56
5.2	An example image from English COCO dataset	57
5.3	An example image from Italian COCO dataset	58
5.4	An example image from Spanish COCO dataset	58

5.5	Distribution of Caption Length in English MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 5 words to a maximum of 49 words	59
5.6	Distribution of Caption Length in Italian MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 6 words to to a maximum of 55 words	59
5.7	Distribution of Caption Length in Italian MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 2 words to to a maximum of 59 words	59
5.8	Graph depicting evaluation loss for pre-trained MarianNMT model with 1x1 convolution on the English MS-COCO dataset	68
5.9	Graph depicting evaluation loss for pre-trained MarianNMT model on the English MS-COCO dataset	68
5.10	Analyzing CNN encoder-modified pretrained MarianNMT transformer: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	70
5.11	Analyzing CNN encoder-pretrained MarianNMT transformer: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	71
5.12	Analyzing CNN encoder-pretrained MarianNMT Transformer for Single model approach: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	73
5.13	Analyzing CNN encoder-petrained MarianNMT Transformer for 1-1 model approach: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	74
5.14	Analyzing CNN encoder-pretrained MarianNMT Transformer for M2 architecture: Saliency map depicting the model's inference on 2000 samples Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	75
5.15	Analyzing CNN-modified Pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	77
5.16	Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	78
5.17	Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	79

5.18	Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	80
5.19	Analyzing CNN encoder-pretrained Marian NMT transformer decoder for Single model using Italian translation of English Dataset: Saliency map depicting the model's inference on translated dataset, offering a visual representation of attention and focus areas in caption generation	81
5.20	Analyzing CNN encoder-pretrained Marian NMT transformer decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	82
A.1	Analyzing CNN encoder-LSTM decoder for Single model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	95
A.2	Analyzing CNN encoder-LSTM decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	96
A.3	Saliency map of a CNN-TransformAnalyzing CNN encoder-transformer decoder for Single model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	97
A.4	Analyzing CNN encoder-transformer decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation .	98
A.5	Analyzing CNN encode-modified pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	99
A.6	Analyzing CNN encoder-modified pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	100
A.7	Analyzing CNN encoder-pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	101
A.8	Analyzing CNN encoder-pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	102
A.9	Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	103

A.10 Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	104
A.11 Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	105
A.12 Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	106
A.13 Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	107
A.14 Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation . .	108
A.15 Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	109
A.16 Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	110
A.17 Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	111
A.18 Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	112
A.19 Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	113
A.20 Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	114
A.21 Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	115

A.22 Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model’s inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation	116
--	-----

List of Tables

4.1	List of the technological components used in this study, supplemented by remarks highlighting their usage and importance	51
5.1	Data Overview: MSCOCO-it Dataset	55
5.2	Data Overview: MSCOCO-es Dataset	56
5.3	Summary of Images and Annotations Statistics: displaying the number of images and corresponding captions for various languages (English, Italian and Spanish)	57
5.4	Performance Metrics English caption Generation by a single model: A Comparative Analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various models	64
5.5	Performance Metrics for Italian caption Generation by a single model: A Comparative Analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various models	65
5.6	Performance Metrics for English caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models	66
5.7	Performance Metrics for Italian caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models	66
5.8	Performance Metrics for Spanish caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models	67
5.9	The above table illustrates the number of times the model correctly/incorrectly generated the caption when it was explicitly instructed about the target language	67
5.10	Performance Metrics comparing the performance between Single model, 1-1 model and M2 model: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.	72
5.11	Performance Metrics comparing the Few-shot perform on various sample size using CNN encoder-Marian NMT decoder: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.	76

5.12	Performance Metrics comparing the Few-shot perform on various sample size using CNN encoder- modified Marian NMT decoder: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.	76
5.13	Performance Metrics for Italian caption Generation using translate-train (translate the English data to Italian using NMT model) and translate-test methods (translate English inference captions to Italian using NMT model): A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr Scores Across Various Models	79

Chapter 1

Introduction

In the current decade, deep learning research has witnessed a pivotal breakthrough, with the initiation led by AlexNet [51]. This Convolutional Neural Network (CNN) secured a remarkable victory in the 2012 ImageNet [66] contest. The versatility of deep networks has been demonstrated across various domains, showcasing their prowess in tasks such as image classification, feature extraction, machine translation, and natural language processing. These networks exhibit the capacity to extract rich information from diverse sources, encompassing audio, images, videos, and text. Among these emerging opportunities, a particular advancement that attracted significant interest was image captioning, which is the primary focus of the thesis.

As humans, we possess the ability to articulate an organized and structured sentence that accurately describes a given image by identifying the prominent content and their relationship with the surroundings. However, describing the content of an image in simple words is a significant challenge for machines. The task requires extracting the clear idea conveyed by the image and training the language model to produce logical and grammatically accurate sentences. The process of discovering an efficient method to analyze an input image, describe its content, and convert it into a series of words, establishing an optimal connection between visual and textual components while ensuring the smoothness of the language, is referred to as "image caption generation." Image captioning is a fundamental problem in artificial intelligence that fuses the knowledge of computer vision and natural language processing. The applications of image captioning are wide-ranging and substantial, including visually impaired individuals [58], autonomous vehicles [35], social media [69], virtual assistants, and more.

1.1 Motivation

Image captioning has made significant advancements in recent years, emerging as a captivating and challenging subject. However, every advancing technology brings forth its own set of challenges. Traditional captioning systems [78] need more naturalness as they generate captions sequentially (i.e., the next word is generated depending on the previous word and the image features), which can lead to semantically irrelevant



Figure 1.1: An example for image captioning task (Source: [7])

language structures. Another limitation is the fact that the majority of research and practical implementations are concentrated mainly on English captioning systems, primarily because of the availability of annotated image-caption datasets in English. Nevertheless, the advantages of cutting-edge technology are not accessible to a significant portion of individuals who do not understand English, given that there are over 7,151 distinct languages spoken globally [4]. Unfortunately, there is a limitation of image-caption datasets in languages other than English. A straightforward method for gathering data in target languages involves using a translation model to translate captions. However, this method has its drawbacks. Notably, it results in a loss of valuable information from the image and relies solely on the language model. As illustrated in Fig. 1.1 the human annotator has described the image as "a train traveling down a track next to a forest." However, upon closer observation, it becomes clear that certain complex details captured visually, such as the distinctive yellow and blue colors of the train, and traversal across three railway lines, are not explicitly conveyed in the caption. The use of a translation model to convert captions into a target language raises concerns about potential loss of such minute information. Therefore, the reliance on English captions as an intermediary introduces a bottleneck in the process. If the original English captions are poorly annotated by human annotators, utilizing a translation model may inadvertently propagate inaccuracies into the translated annotations in the target language. To address this, there is a need for a multi-modal systems that seamlessly integrate both visual and textual descriptions, ensuring a more comprehensive representation. Hence, we utilize a technique to address the language barrier by introducing cross-lingual transfer learning in image captioning.

Cross-lingual image captioning generates accurate and fluent captions in the target language. For instance, let's consider the scenario where the objective is to provide a description for an image in the Spanish, but there is a limitation for image-caption data in Spanish. So, the model is initially trained with English labels, and then applying the knowledge it has gained to generate Spanish captions. Hence, the process of transmitting image data across several languages is referred to as cross-lingual image captioning. Significant advancements have been made in this subject in recent years. In addition, researchers have devised strategies to address the linguistic obstacles in image captioning tasks, such as manually annotating image-caption datasets in a certain language [61], unpaired image captioning [34], and employing transfer learning [19] approaches.

In the thesis, we investigate a method for cross-lingual transfer learning for image captioning. This technique is based on an existing technology that was originally designed for translating text in multiple languages by Lyu et.al. [57]. The objective is to redesign the multilingual model for image captioning task. Therefore, in the study we conduct a

comprehensive evaluation of three frameworks: the Single model [[78], [84], [57]], 1-1 model [[75], [57]], and M2 [57] model, each designed for multilingual image captioning. The Single model adopts a technique of employing several unidirectional models, which is found to be successful when there is a significant amount of data available in the desired language. In contrast, the 1-1 model uses a single encoder and decoder to generate captions for multiple languages. Despite it sharing the model parameters across multiple languages, this system faces a trade-off between the number of languages introduced and the quality of the captions. Finally, the M2 model introduces a unique approach by sharing only language-specific modules. The M2 model utilizes a modularized architecture to establish an inter-lingual space, which enables convenient and efficient modifications to the model. This design choice allows for flexibility and adaptability, showcasing the M2 architecture as a promising solution for multilingual image captioning tasks.

1.2 Research problem

The existing literature predominantly overlooks the cross-lingual aspect of image captioning, hindering its potential impact on a global scale. In the thesis, we address the critical research problem of developing a method for cross-lingual multi-modal task, with a specific focus on image captioning. To tackle the challenge, we draw inspiration from a well established approach initially designed for multilingual machine translation [57] and adapt the methodology for cross-lingual image captioning. Therefore, the research challenge focuses on creating a robust yet adaptable framework that bridges the language gap and enable image understanding and description in multiple languages. The study focuses on proposing a novel approach that can transfer knowledge from high-resource language to low-resource language (enabling few-shot learning).

Main research question: Can adapting the M2 model to the image captioning task enable few-shot captioning? To what extent is the model able to generate meaningful captions (in a few-shot setup)?

1.3 Research objectives

The central focus of the study is to address the challenges arising from scarcity of data in non-English languages. The key objective is to foster effective cross-lingual communication by implementing a system to automatically generate captions in the target (non-English) language. The approach aims to expand the range of languages supported by the captioning system, to promote a framework that is more inclusive and accessible, surpassing linguistic limitations. As a result, we redesign the M2 architecture to perform cross-lingual image-captioning tailored for low-resource languages in a few-shot learning context.

The secondary objective of the study is to conduct comprehensive comparative analysis between three approaches, such as single model, 1-1 model, and M2 model. Additionally, our objective is to analyze the distinctions among different decoders, with a specific focus on the LSTM [39], Transformer [76], and pre-trained Transformer. This comparative study is crucial in providing insights into the subtle ways in which different approaches generate captions. Our objective is to thoroughly assess the performance and attributes of distinct frameworks in order to enhance our understanding of their individual strengths and shortcomings. This will provide useful insights into the wider field of multilingual

image captioning.

Having established the key research objectives and centered the main research question on the core concept, we proceed to articulate research questions that directly address the design, implementation, and logic of our approach.

Research question:

1. **How does the quality of generated captions differ among the single model, 1-1 model, and M2 model?**
2. **How does performance differ across different decoder architectures: LSTM, Transformer, and Pre-trained NMT Transformer?**
3. **Can adapting the M2 model to the image captioning task enable few-shot image captioning?**
4. **To what extent is the model able to generate meaningful image captions (in a few-shot setup)?**

1.4 Research methodologies

To achieve proficient multilingual image captioning, it is crucial to employ a thorough and methodical research methodology that integrates techniques from both computer vision and natural language processing. The methodology expands upon a modular approach that was first suggested for multilingual Neural Machine Translation (NMT) tasks [57], adapting it for the purpose of cross-lingual image captioning. We implement three distinct models. The initial approach involves employing multiple single-directional models [[78], [84]], wherein each model possesses an encoder for image input and a decoder for processing captions in the target language. However, this strategy becomes impractical due to the quadratic increase in the number of models as additional languages are introduced [57]. The second method, adopting a 1-1 configuration that restricts the number of models by sharing parameters. This method utilizes a single encoder and a single decoder for captioning in multiple languages, resulting in a compact structure that reduces the overall number of parameters. The 1-1 model encounters a capacity bottleneck, manifested when the model is constrained by the trade-off between the number of languages introduced and captioning accuracy. The final method M2, not only streamlines the system but also proves to be effective in enhancing performance by incorporating language-specific decoder. Therefore, the core concept of this study revolves around the implementation of cross-lingual transfer learning approach using the M2 model. By doing this we harness knowledge acquired from high-resource languages to enhance the performance of model in low-resource languages. To measure the performance of a model, it is important to employ well-established metrics like BLEU [62], ROUGE [54], and CIDEr[77]. These metrics ensure a thorough examination across different languages and provide an accurate evaluation of the model’s effectiveness. The experiment’s code has been made available.

1.5 Scope and limitations

The primary focus of this research lies in designing customized models for multilingual image captioning, a technology that aims to overcome language barriers and enhance

multi-modal communication. The models strive to improve the user experience by creating captions in the desired language, facilitating effortless sharing and understanding of visual content among people with different linguistic backgrounds. The efficacy of the implemented framework is intricately tied to the availability and quality of sparsely annotated image-caption pairs across diverse languages.

Nevertheless, it is crucial to recognize the inherent limitations associated with the task at hand. Firstly, the computational requirements for the system are substantial, demanding increased computational power, typically facilitated by GPU. Also, to effectively perform few-shot learning in a cross-lingual context, a prerequisite is having at least a small sample of data in the target languages. This poses a potential limitation, especially for languages with scarce resources.

1.6 Contribution

The following contributions are made in this thesis:

1. Redesigned the modular M2 approach, originally designed for Machine Translation tasks, to address challenges encountered in Multilingual Image Captioning.
2. Customized the pretrained Marian NMT Decoder from Hugging Face by introducing 1D (1×1) Convolution layers to perform Cross-Attention [76] between Feature Maps from Image Encoder & Causal Word Embeddings of Text Decoder.
3. Evaluated the M2 architecture under different few-shot scenarios. This involved evaluating the performance and adaptability of the model over varying sizes of training data sets, thus providing useful insights into its robustness and ability to generalize.
4. Implemented a unified architecture (1-1 model), specifically for multilingual image captioning tasks. This model was concurrently trained on datasets from multiple languages, such as English (en), Italian (it), and Spanish (es), allowing it to acquire knowledge and produce descriptions in a multilingual setting.
5. Performed separate evaluations for each language incorporated into the unified 1-1 model. Leveraging the Python Lang_detect library [8] and the generative pre-trained transformer model[22], language detection mechanisms were employed. This involved verifying whether the captions generated by the model aligned accurately with the target language.
6. Improved cross-modal grounding and prevent the model from exploiting language priors [87]. This improvement aims to strengthen the model's ability to establish meaningful connections between visual and textual elements, ensuring that its predictions are based on intrinsic visual features rather than relying excessively on pre-existing language biases.
7. Performed ablation experiments using different auto-regressive language model architectures such as Bi-LSTM [33] and Transformer [76].

1.7 Outline of the thesis

The outline of the report is as follows:

1. **Chapter 2:** This section provides a comprehensive overview, explaining fundamental concepts and introducing key ideas, theories, and terminology that are essential to the study. This guarantees an in-depth understanding of essential topics, which include deep learning architectures, image processing techniques, and natural language processing.
2. **Chapter 3:** The study conducts a thorough review of both historical and current literature concerning the topic at hand. The chapter investigates different methodologies and addresses the challenges associated with generating significant captions. We also examine pertinent literature, emphasizing the modifications and effects observed in specific studies that have helped shape our research.
3. **Chapter 4:** This section provides a detailed analysis of the methodology and execution complexities, exploring key elements to reveal insights into their formulations, motivations, and underlying principles. The chapter concludes with a thorough clarification of the training methodologies and implementation details.
4. **Chapter 5:** This chapter starts with a detailed overview of the datasets and benchmarks used, followed by an analysis of the specific evaluation criteria. We conduct a comprehensive assessment of our model's performance across multiple benchmarks, comparing it with both baseline and state-of-the-art techniques. The chapter presents both quantitative and qualitative evaluations of our strategy.
5. **Chapter 6:** Provides a brief summary and conclusion to the thesis, including the limitations and potential areas for future research.

Chapter 2

Concept

This chapter aims to establish the fundamental basis required for understanding the complex concepts, theories, and procedures that form the foundation of our research. The initial step is clarifying essential ideas related to Image Captioning. Then, we give a perceptive summary of the current approaches and thoroughly analyze the underlying issues. Subsequently, we proceed to explore the fundamental concept of our proposed study and offer a justification for its basis.

2.1 Preliminaries

In this section, our attention is directed towards the core ideas essential for grasping the concepts addressed in the thesis. We begin by examining the fundamental principles of Artificial Intelligence (AI) and its subdomain, Machine Learning (ML). Next, we explore specific intricacies regarding different architectures and concepts employed in Deep Learning (DL), as well as its influence on image captioning.

2.1.1 Artificial Intelligence (AI) and Machine Learning (ML)

Artificial Intelligence (AI) is a prominent field in Computer Science that focuses on the research and creation of intelligent algorithms, techniques, and systems. The primary objective is to enhance the capabilities of robots to perceive and understand their environment, acquire knowledge from past experiences, and make intelligent decisions to maximize their likelihood of achieving desired outcomes. AI aims to replicate several aspects of human cognitive processes, including visual and auditory perception, language understanding and generation, data processing, recommendation systems, and other related functions. The basis of AI is rooted in the deliberate implementation of logic and decision trees, enabling machines to possess independent learning, logical thinking, and self-correcting abilities.

The evolution of AI has experienced a significant shift, transitioning from its initial focus on symbolic logic-based systems to adopting data-driven methodologies that

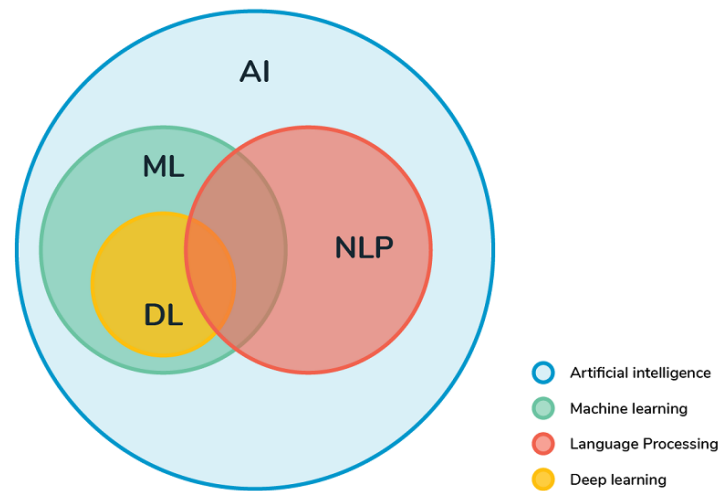


Figure 2.1: Relationship between Artificial Intelligence, Machine Learning, Deep Learning and Natural Language Processing (Source: [60])

are integral to Machine Learning techniques. Symbolic AI, based on the manipulation and interpretation of symbols that represent things or concepts, mainly relies on logic-based programming. This programming approach involves the utilization of rules and axioms to make inferences and deductions. Symbolic AI encountered difficulties in effectively negotiating complex and ever-changing real-world situations, despite its notable advantages. This led to a significant change in the field, resulting in the emergence of specialized sub-fields such as Machine Learning, Deep Learning, Robotics, and Natural Language Processing (as depicted in Fig.2.1). These advancements have collectively propelled AI to become a powerful force that is expected to transform the technological landscape. The following sections will explore the intricacies of these sub-fields, revealing their distinct contributions to the constantly expanding landscape of AI.

Machine Learning (ML) is a specialized area within AI, characterized by its ability to adjust and respond to new information or circumstances. ML algorithms exhibit an innate capacity to identify patterns and derive meaningful conclusions from data, distinguishing them from conventional AI techniques. This allows them to break free from strict adherence to predefined rules, resulting in a significant advancement. This evolution is a crucial turning point, laying the foundation for the current era of AI. The primary objective of machine learning is to create computer algorithms that can independently acquire and integrate knowledge gathered from data. This learning process occurs through a systematic analysis of observations, which may include personal experiences, examples, and explicit instructions. The objective is to uncover fundamental patterns that empower the system to autonomously make judgments without any requirement for human intervention. ML exerts a substantial impact on diverse sectors such as healthcare, banking, e-commerce, and autonomous vehicles, resulting in profound and transformative advancements. Applications have diverse purposes, such as improving personalized recommendation systems and developing medical algorithms for disease identification.

ML covers a wide range of learning methods, each relying on factors such as having carefully curated datasets, the adequacy of sample sizes for training models, the presence of pre-trained models, and the computational resources available. This introduction lays the groundwork for a comprehensive analysis of the various aspects of ML, elucidating its distinct methodologies and applications within the overarching framework of Image Captioning.

The domain of ML includes various learning methodologies, each tailored to address certain challenges and tasks. Three frequently employed strategies comprise:

1. **Supervised Learning:** This method entails utilizing a dataset that has been annotated with labels in order to train an algorithm. Supervised learning involves a training dataset that comprises pairings of input and output. The inputs are the features, and the outputs are the associated labels. The main objective is to acquire a mapping function that can effectively forecast outputs for new and unfamiliar inputs. Supervised learning is commonly used for tasks such as classification, audio recognition, and regression analysis.
2. **Unsupervised Learning:** In contrast, unsupervised learning utilizes datasets that do not possess explicit labels or annotations. This methodology functions without explicitly specified outputs and seeks to reveal hidden patterns, structures, or relationships within the data. Unsupervised learning is advantageous for examining the inherent structure of data or carrying out tasks such as grouping, dimensionality reduction, and density estimation.
3. **Transfer Learning:** The transfer learning paradigm involves reusing a model that has been trained on one task for a different, yet related, task. This strategy utilizes the acquired information from the source task to enhance the learning of the target task, particularly in situations when there is a scarcity of data for the target task. It has proven valuable across various domains, such as image recognition, natural language processing, and other complex tasks, enabling models to benefit from previously acquired knowledge and adapt to new challenges.

Apart from these fundamental approaches, there exist various other learning strategies, including reinforcement learning, self-supervised learning, semi-supervised learning, multi-instance learning, and others. In this thesis, we adopt a hybrid approach that combines the advantages of supervised learning and transfer learning methods.

2.1.2 Deep Learning (DL)

Deep learning is a distinct branch of ML that presents its own unique set of opportunities and challenges. It stands out due to its exceptional ability to extract meaningful patterns and representations from data sources, such as images, videos, and text, among others. Remarkably, DL does not rely on pre-existing human skills or domain-specific information. The term "deep" refers to the multiple layers of neural networks that are employed to detect complex patterns in data. The main goal of Artificial Neural Networks (ANN) is to imitate the cognitive abilities of the human brain, although on a smaller scale, by collecting knowledge from large datasets. DL architectures have a distinctive ability to swiftly process raw input, akin to the functioning of the human brain, and then improve their predictive accuracy as the amount of data increases. DL plays a crucial role in achieving high levels of precision and accuracy in several tasks, including speech

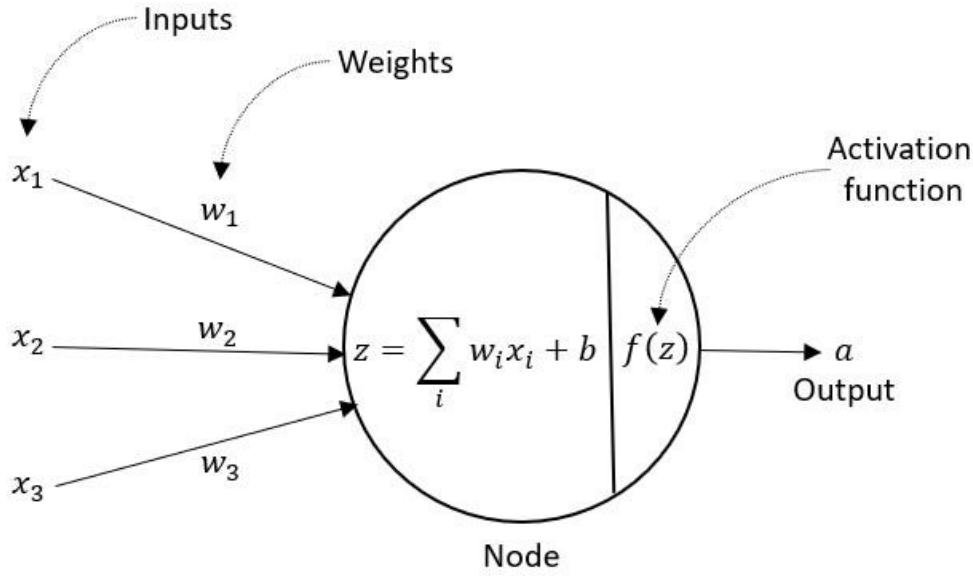


Figure 2.2: Schematic representation detailing the structure of a single perceptron, showcasing its neuron and connections in the neural network framework (Source:)

recognition, translation, and object detection. The advancements in AI have been seen through notable achievements such as Google DeepMind's AlphaGo, intelligent voice assistants, self-driving cars, and other key discoveries.

Deep Learning concepts:

The perceptron [65] is a fundamental building block and serves as the primary computational unit within a Neural Network (NN). Frank Rosenblatt introduced the concept in 1957, and it is the most basic type of NN, specifically tailored for binary classification problems. This approach utilizes supervised machine learning and exploits a linear decision boundary, known as a hyperplane, to effectively separate and classify input data. The perceptron, despite its simplicity, has the ability to handle difficult categorization problems by learning and adapting to the input it receives, much like the intricate neuronal connections in our brains.

Let break down the individual components of a fundamental perceptron employing a neuron, as illustrated in Fig. 2.2;

1. **Input layer:** The neuron receives input in the form of numerically encoded data, denoted as $(x_1, x_2, x_3, \dots, x_n)$, alongside their corresponding class labels $y_1, y_2, y_3, \dots, y_n$
2. **Weighted summation:** The inputs $(x_1, x_2, x_3, \dots, x_n)$, undergo a transformation by being multiplied with corresponding weights $w_1, w_2, w_3, \dots, w_n$, initialized with random values. Additionally, a constant weight or bias (b) is incorporated, contributing to the weighted sum s ,

$$s = (w_i^T x_i + b) \quad (2.1)$$

3. **Activation Function:** The weighted sum, denoted as s , is passed via a non-linear

activation function σ , such as Sigmoid, Tanh, or Rectified Linear Unit (ReLU) [15]. The activation function incorporates non-linear properties and regulates the range of the output. Specifically, the use of ReLU ensures that the resulting output a is confined to the interval from zero to positive infinity.

$$a = \sigma(s) \quad (2.2)$$

4. **Output and Learning:** The final result a represents the altered input data. Subsequently, the accuracy of this output is assessed. The Back Propagation approach is used iteratively to enhance the neuron's forecasting power.

The single layer perceptron is limited in its ability to linearly distinguish between diverse input data distributions, as it only depends on a single linear hyperplane. To overcome this limitation and effectively capture complex non-linear patterns, a more advanced architecture with multiple layers of perceptrons is required. This architecture is referred as a Multi-Layer Perceptron (MLP) or Deep Neural Network (DNN). Nevertheless, when linear functions are intricately combined, the total model operates as a single layer feed-forward model without the incorporation of non-linear elements. As a result, our model has difficulties in accurately representing complex patterns within the dataset. To address this problem effectively, it is essential to utilize non-linear activation functions during the forward pass. Therefore, the neural network training procedure comprises two critical stages:

1. **Forward Propagation:** The operations performed in this phase are similarly to the training processes discussed earlier in the perceptron section.
2. **Backward Propagation:**
 - (a) **Gradient calculation:** The process begins by calculating the gradients of the loss function in relation to the model parameters.
 - (b) **Weight Update:** Gradients are utilized to modify both the biases and weights of the network in order to minimize the loss.
 - (c) **Backward pass:** The process of back-propagation involves the sequential transmission of gradients through the network, from one layer to another, by applying the chain rule of calculus.
 - (d) **Optimization Algorithm:** Weight adjustments are guided by optimization algorithms such as gradient descent to effectively minimize the loss.
 - (e) **Learning rate adjustment:** The learning rate, which is a hyper-parameter, affects the magnitude of weight updates and has an effect on the model's convergence.

The Forward and Backward passes are iteratively performed within a loop until the model reaches a global optimum or the loss function shows a significant decrease.

2.1.3 Convolution Neural Networks (CNN)

The Convolutional Neural Network (CNN) proposed by LeCun et al. [?] has played a significant role in the recent advancement of Deep Learning, particularly for image analysis. CNN, sometimes referred to as ConvNet, is a specialized neural network

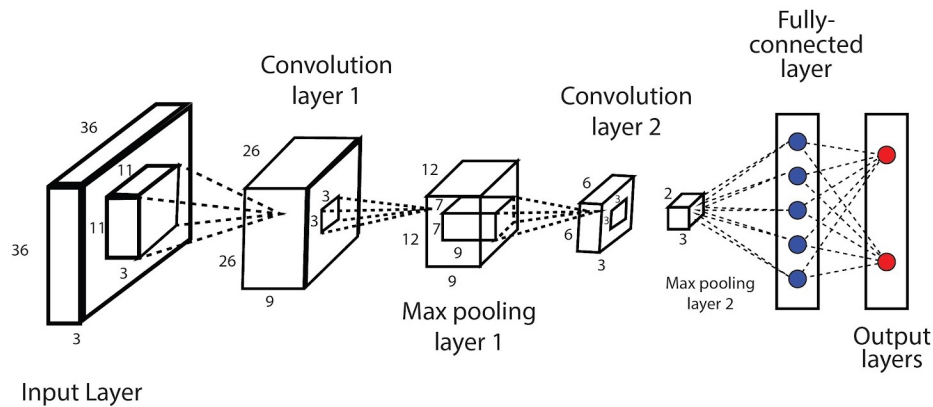


Figure 2.3: A CNN is made up of two primary parts. The process of feature learning and the subsequent classification component. In the feature learning phase, a series of convolutional layers are sequentially arranged to acquire knowledge of features, beginning with fundamental ones such as edges and progressing towards complex ones. The classification layer is comprised of a series of fully connected (FC) layers that allocate the input to specific classes (Source: [1]).

specifically developed for the examination and manipulation of data structured in a grid-like arrangement, such as images. A digital image is a form of data that uses a binary system to describe visual information. It is structured as an array of pixels, where every pixel contains distinct values that indicate its brightness and color.

The human brain demonstrates fast information processing in response to visual stimuli. Every single neuron functions within its own receptive area and establishes connections with other neurons to collectively cover the full visual field. Just like individual neurons in the biological vision system, each neuron in a CNN solely processes input within its own receptive field. The layers are arranged in a hierarchical fashion, where the earliest levels emphasize the recognition of basic patterns like lines and curves, while more complex patterns like faces and objects are learned in the following layers. CNNs consist of multiple layers, each serving a specific purpose in the extraction of features from input data. The conventional layers commonly present in a CNN (fig.2.3) consist of

1. **Input Layer:** The initial layer is responsible for receiving the input data, typically in the form of an image. Every node inside this layer corresponds to either a pixel or a characteristic of the input.
2. **Convolution Layers:** The CNN relies on the convolution layer as its foundational element, which carries the majority of the computational load. Sparse interaction, equivariant representation, and parameter sharing are the three main advantages of convolution in computer vision.

The layer conducts matrix computation by evaluating the dot product of two matrices. The kernel matrix contains modifiable parameters, whereas the other matrix represents a limited region inside the receptive field. The kernel captures the intricate details, despite having a smaller spatial dimension compared to the image. As illustrated in fig.2.4, consider an image with three channels (RGB). The dimension of the kernel is relatively smaller in terms of height and width but can cover all the channels in terms of depth. During the forward pass, the kernel moves

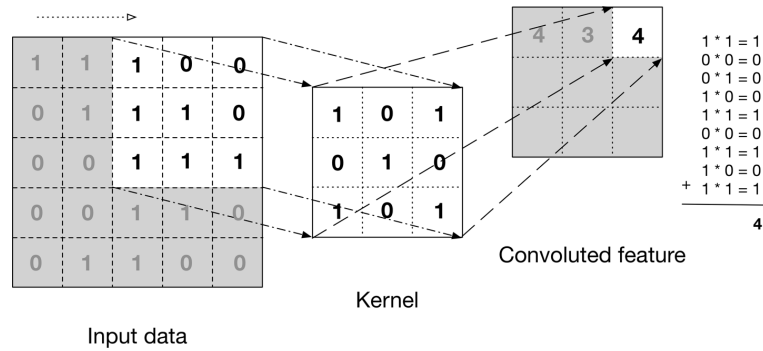


Figure 2.4: Illustration of feature extraction using convolutional filters or kernels (Source: [3])

across the height and width of the image, generating a representation for each receptive region. The procedure yields a two-dimensional activation map, which offers details regarding the kernel's reaction at each spatial location within the image. The variable size of the kernel during its traversal across the input is known as stride.

3. **Pooling (Sub-sampling or Down-sampling) Layers:** The main objective of this layer is to reduce the size of the convolved feature map, with the aim of minimizing computational expenses. This is achieved by reducing interconnections between layers and performing operations on individual feature maps. There are different pooling operations, such as max pooling (identifies the highest value within a receptive field) [21] and average pooling (computes the mean value within the field) [21]. Although pooling results in a loss of knowledge, it offers various benefits, including complexity reduction, improved efficiency, and prevents over-fitting.
4. **Fully Connected (Dense) Layers:** Integrating a Fully-Connected layer is widely recognized as a cost-efficient method for gaining insights into non-linear combinations of the high-level characteristics, which are expressed by the output of the convolutional layer. The fully connected layer is learning the intricacies of a potentially non-linear function inside the specific domain. Following the conversion of the input image into a format compatible with the Multi-Layer Perceptron, the subsequent step entails transforming the image into a column vector by flattening it. The compressed result is fed into a feed-forward neural network, and the back-propagation algorithm is utilized during each iteration of the training procedure.
5. **Output Layer:** The final layer is responsible for producing the network's output. The composition of this layer varies depending on the specific task being performed. Nodes that represent multiple classes can be utilized in image classification, employing a softmax activation function to generate a probability distribution.

CNN Variants:

There are four popular variants of Convolutional Neural Networks (CNNs) that have become widely used in imagine captioning research. The computer vision field broadly recognizes the following CNN architectures: AlexNet, GoogLeNet, ResNet, and VGGNet.

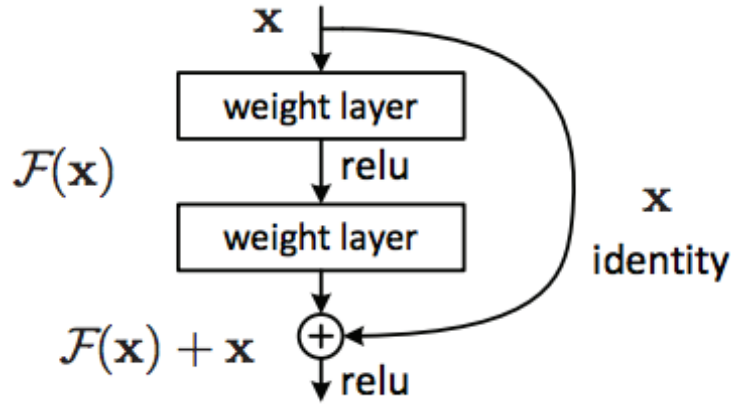


Figure 2.5: The concept of residual connection is employed in the ResNet architecture, as described in the work by ABC. The fundamental concept of residual learning pertains to the optimization of a stack of layers by learning a residual mapping $F(x)$ instead of the original mapping $H(x)$. This is achieved by expressing $H(x)$ as the sum of $F(x)$ and x (Source: [11])

The AlexNet architecture, created by Krizhevsky et al. (2012) [?], played a crucial role in stimulating the rise of deep learning by obtaining a definitive triumph in the 2012 ILSVRC competition. During the next two years, the ILSVRC-2014 competition observed the victory of a unique form of CNN referred to as GoogLeNet [72], which was created by Google. In the same year, another highly acclaimed architectural design called VGGNet [70] was also introduced. The ILSVRC-2015 competition saw the introduction of Residual Network or ResNet [37] architectures, which is notable for its substantial depth, with up to 256 layers.

ResNet Architecture:

Drawing insights from empirical evidence in experimental studies [37], we explore the challenges linked to training highly deep neural networks, exemplified by architectures like VGGNet [70]. As neural networks become deeper, they often encounter difficulties such as vanishing gradients and degradation in training accuracy. The obstacles outlined impede the effective training of networks, constraining their ability to capture and learn complex hierarchical features. The transformative innovation of ResNet lies in its adoption of residual blocks, overcoming issues of vanishing gradients or degradation and enabling the successful training of exceptionally deep networks.

Given the strong capability of neural networks as function approximators, it is reasonable to expect that they would possess the ability to effectively address the identification function, wherein the output of a function corresponds to its input. The fundamental component of ResNet is the residual block (as illustrated in Fig.2.5). Consider x as the input to the block, and $F(x)$ as the intended underlying mapping. The resulting value y of the residual block is acquired through [37]:

$$y = F(x) + x \quad (2.3)$$

To facilitate the learning of $F(x)$, instead of directly learning the mapping, the block

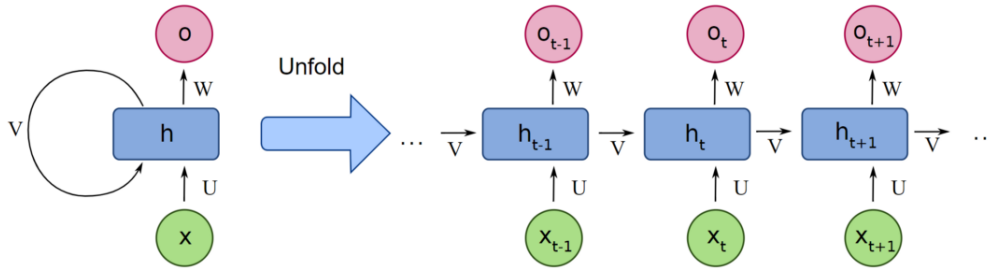


Figure 2.6: A representation of an RNN is shown on the left side, and an RNN that has been unfolded (or unrolled) into a full network is shown on the right side. The term "unrolling" refers to the process of representing the network by explicitly writing out its structure for the entire series. For instance, in the case when the sequence of interest is a sentence consisting of three words, the neural network would be expanded into a network with three layers, with each layer corresponding to a specific word. (Source: [2])

learns the residual mapping $F(x)x$. The shortcut connection allows the gradient to flow through the identity mapping if $F(x)$ is close to zero during training, thus eliminating the vanishing gradient problem.

The ResNet architecture is constructed by stacking numerous residual blocks, featuring a fundamental structure that incorporates convolutional layers, batch normalization, ReLU activation functions, and residual connections. To optimize computational efficiency, ResNet commonly employs bottleneck blocks, which consist of three sequential convolutions: 1×1 , 3×3 , and another 1×1 . Additionally, the architecture incorporates down-sampling layers designed to decrease spatial dimensions.

2.1.4 Recurrent Neural Networks (RNN)

Recurrent neural networks (RNNs) are designed specifically to handle sequential input data. They excel particularly in tasks that require the examination of temporal correlations, such as natural language processing and image captioning. RNNs, in contrast to conventional feed-forward neural networks, possess a distinctive architecture that enables the preservation of information from previous inputs via hidden states. RNNs include an inbuilt memory mechanism that enables them to effectively collect and utilize sequential patterns, making them well-suited for jobs that demand contextual understanding. They are adept at producing meaningful outputs by taking into account the sequential organization of data, thus acknowledging the importance of input order.

A basic RNN, depicted in Figure 2.6, is commonly constructed by combining the result from the previous time step with the input from the current time step. The presence of recurrent connections allows them to effectively store and recall information over temporal data. RNNs utilize the hyperbolic tangent (\tanh) activation function in the hidden layer to improve the network's ability to capture long-term dependencies. As depicted in fig. 2.6, 'o' is denoted as the output variable and is considered to provide non-standardized logarithmic probabilities for each possible value of the discrete variable. In order to obtain a probability distribution throughout the output, a common post-processing step is using the softmax operation. This method standardizes the non-standardized logarithmic probabilities, providing a vector θ that represents the model's prediction as standardized probabilities for each output.

Back-propagation is essential for training, as it enables the learning of sequential relationships and the optimization of network parameters. Gradients are computed in the back-propagation process by temporally propagating errors through the network. The process involves extending the network across input sequences and calculating gradients at each individual time step. The frequently used term for this approach is back-propagation through time (BPTT) [81]. The gradients at each output are interconnected due to the shared parameters across all time steps, relying on both current computations and previous steps. While the typical technique proves to be beneficial, it encounters challenges such as vanishing gradients [38] and exploding gradients. These issues arise when the gradients decrease or increase excessively over lengthy sequences, which impairs efficient learning [38]. To address these issues, advanced architectures such as Long Short-Term Memory (LSTM) [39] have been proposed. LSTM enhances the capacity of RNNs to effectively capture and retain long-term relationships.

Long Short-Term Memory (LSTM)

LSTM networks, a particular variant of RNNs, are renowned for their exceptional ability to capture and model long-term dependencies in data. LSTMs, first introduced by Hochreiter and Schmidhuber in 1997 [39], have proven to be highly effective in various applications by addressing the issue of vanishing gradients which is the major limitation of traditional RNNs. The LSTM model, illustrated in Figure 2.7, exhibits remarkable adaptability for the classification, manipulation, and forecasting of time series data, particularly when faced with temporal lags of uncertain duration. LSTMs are improved with the integration of memory cells and gating mechanisms, enabling them to selectively store and forget information. The system consists of several essential components, including the cell state, an input gate, a forget gate, an output gate, and a set of weights and biases. We will analyze the operation of an LSTM cell in a methodical manner, reviewing each step separately.

1. **Forget gate f_t :** The forget gate is responsible for determining the retention or discarding of information from the previous cell state C_{t-1} . The forget gate activation vector is generated using the previous hidden state h_{t-1} and the current input x_t as input values.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

In this case, W_f and b_f represent the weight matrix and bias for the forget gate, respectively, while σ denotes the sigmoid activation function.

2. **Input gate i_t and Candidate Cell State \tilde{C}_t :** The input gate determines the selection of new data to be stored in the cell state. The input gate, similar to the forget gate, receives h_{t-1} and x_t as inputs and produces an activation vector for the input gate. Moreover, it calculates a candidate cell state \tilde{C}_t that represents the new data to be included.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.5)$$

where, W_i, W_c, b_i and b_c are the weights and biases for the input gate and the candidate cell state, respectively.

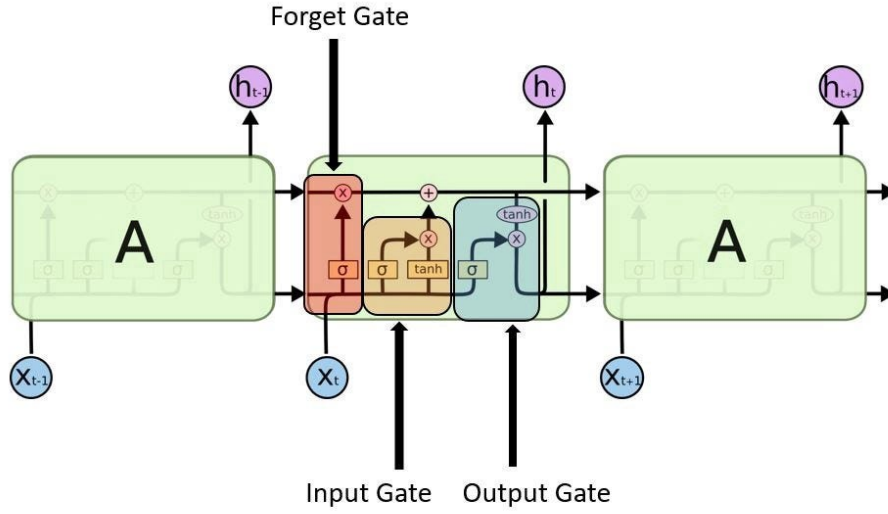


Figure 2.7: An LSTM Block contains four interacting layers (cell state, an input gate, a forget gate, an output gate) (Source:[14])

3. **Cell state C_t :** The cell state is updated by combining the previous cell state C_{t-1} , the forget gate f_t , and the input gate multiplied by the candidate cell state $i_t \cdot \tilde{C}_t$.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (2.6)$$

4. **Output gate o_t and Hidden State h_t :** The output gate calculates the next hidden state h_t using the modified cell state. The function accepts two inputs, h_{t-1} and x_t , and produces an output gate activation vector o_t .

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.7)$$

The hidden state is then calculated as,

$$h_t = o_t * \tanh(C_t) \quad (2.8)$$

Although LSTM networks are highly effective in capturing long-term relationships in sequential data, they come with certain drawback like heavily depending on previous states to generate output. In order to overcome this limitation, researchers have created the bi-directional BLSTM model [33]. This model employs bidirectional processing, enabling it to incorporate historical and future information pertaining to a particular point in a given sequence. Although there have been advancements, it is crucial to recognize that LSTMs, especially BLSTMs, are still susceptible to computational complexity, sensitivity to hyper-parameters, and difficulties in capturing very long-term relationships.

2.1.5 Transformers

The implementation of the Transformer by Vaswani et al. [76] in 2017 marked an important shift in the encoder-decoder framework. The sequential structure of RNNs presented computational constraints, hindering effective parallelization on modern hardware such as Tensor Processing Units (TPUs) and Graphics Processing Units (GPUs). Sequential processing required the word-by-word processing of phrases, hence constraining the practicality of parallel computing. The Transformer model overcame these restrictions by incorporating a groundbreaking technique called the self-attention mechanism.

Self-Attention Mechanism:

The key feature of the Transformer lies in its integration of the attention mechanism. This method is crucial in the word processing stage as it allows the model to selectively focus on other words in the input that have a strong semantic relationship with the word being analyzed. For the purpose of demonstration, let us examine two sentences:

1. The cat drank the milk because it was hungry.
2. The cat drank the milk because it was sweet.

In the initial statement, the pronoun "it" refers to the noun "cat," whereas in the subsequent sentence, it refers to the noun "milk." The self-attention mechanism is essential for providing the model with more contextual information about the word "it" during processing. The model's contextual awareness allows it to dynamically associate "it" with the correct antecedent, whether that be "cat" or "milk." Vaswani et al. (2017) [76] utilize dot-product attention, which can be expressed by the following equation:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2.9)$$

where K and V represent the keys and values, respectively and Q represent the query matrix. The self-attention mechanism involves a series of steps to calculate attention scores and produce a weighted sum, so enhancing the model's ability to understand complex relationships in the input sequence. Below is a detailed explanation of the self-attention calculation in the transformer model.

1. **Key, Query, and Value Transformations:** For each word i in the input sequence, the Transformer computes three vectors: Key K_i , Query Q_i , and Value V_i . These vectors are obtained by linear transformations of the word's embedding vector E_i :

$$K_i = W_K.E_i; Q_i = W_Q.E_i; V_i = W_v.E_i$$

where W_K, W_Q, W_V are learnable weight matrices.

2. **Score Calculation:** Let's assume that we are figuring out the self-attention for the first word in this example, "The". Each word in the input sentence must be evaluated in relation to this word. As we encode a word at a specific position, the score dictates how much attention should be paid to other components of the input. This score is calculated by taking a dot product of query and key matrix of that respective word that is been scored currently.

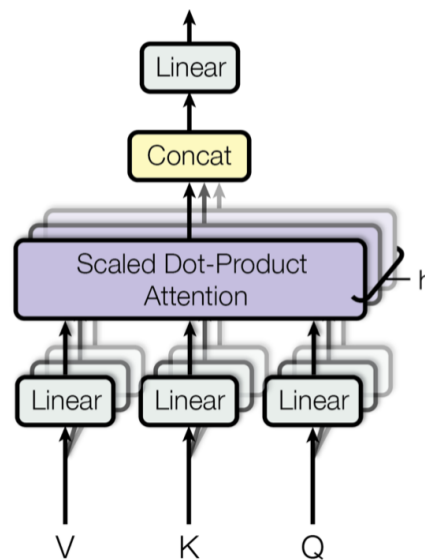


Figure 2.8: Multi-Head Attention with h attention heads, all running in parallel. (Source: [80])

3. **Normalization with Softmax:** The obtained score is then divided by 8 (as the square root of the key vector, i.e., 64, is utilized in the paper to ensure stable gradients). This result is fed into the softmax function, normalizing the scores to be positive and summing to one. The softmax score determines the emphasis each word receives in this context, allowing occasional focus on words connected to the present word.
4. **Weighted Value Matrix:** Subsequently, each value matrix is multiplied by its corresponding softmax score. This process aims to attenuate the influence of irrelevant words while preserving the significance of the word or words targeted for attention.
5. **Output Generation:** Finally, the weighted value matrices are summed, producing the output of the self-attention layer. This output is then fed into the feed-forward network, contributing to the model's ability to capture intricate relationships within the input sequence.

Multi-Head Attention:

Multi-head attention [76] as shown in Fig.2.8, is a crucial element of the Transformer, specifically to improve the model's capacity to comprehend a wide range of complex patterns in input sequences. The self-attention method is executed in parallel across many "heads," each equipped with its own set of trainable weight matrices for key, query, and value. By utilizing several heads, the model is able to simultaneously concentrate on various elements and relationships within the data. Each head acquires distinct representations, enabling the model to effectively capture diverse forms of dependency in the input. The outputs of the individual attention heads are concatenated and then linearly processed to generate the final output. As a result, the transformer architecture

becomes highly versatile and robust for various tasks.

The benefits of multi-head attention includes increased modeling capacity and the capability of covering a wider spectrum of patterns in the data. Moreover, the parallel computation across several heads can result in enhanced efficiency during the training process. Nevertheless, it is important to take into account certain drawbacks. Multi-head attention increases the computational complexity of the model, necessitating a greater number of parameters and resources. This can lead to extended training periods and enhanced memory requirements. Although facing these difficulties, the benefits of multi-head attention make it an excellent tool for strengthening the capabilities of transformer designs.

Positional Encoding:

Positional encoding [76] is another crucial component of transformer that conveys information about the positions of tokens within a sequence. Transformers, unlike RNN or CNN, do not include inherent sequential-order information. This is because transformers analyze the entire sequence in parallel. Positional encoding overcomes this constraint by incorporating positional information into the input embedding. The most common technique for positional encoding involves using sine and cosine functions.

This encoding approach guarantees that tokens at various places are assigned unique positional embedding. By using both sine and cosine functions, the positional encoding mechanism is able to effectively represent independent frequencies. This enables the model to accurately differentiate between tokens located at different distances within the sequence. The sinusoidal nature of each dimension in the positional encoding allows the model to efficiently process longer sequence lengths. As a result, it is feasible to infer the relative placement of different embedding at a relatively low cost. The positional encoding is then included by adding it element-wise to the input embedding of the tokens. The transformer model is provided with a fused representation that integrates both the token and positional information.

The Overall Model Architecture:

The Transformer model comprises of encoder and decoder components, as depicted in Figure.2.9. The encoder and decoder consist of multiple identical layers, which can be stacked N_x times.

Encoder: The encoder's task is to analyze the input sequence and provide a thorough representation that captures the contextual information of each input tokens. The fundamental components of the encoder consist of:

1. **Multi-Head Self-Attention:** This technique enables the model to assign varying weights to distinct segments of the input sequence, thereby capturing intricate connections and interactions within the data.
2. **Position-wise Feed-Forward Networks:** Each layer includes a position-wise fully connected feed-forward network. The network employs the attention mechanism to process information in a position-specific manner, allowing the model to gain understanding of non-linear shifts and complex patterns within the input sequence.
3. **Layer Normalization and Residual Connections:** After each sub-layer, layer normalization is applied, and the output of the sub-layer is connected to the input using a

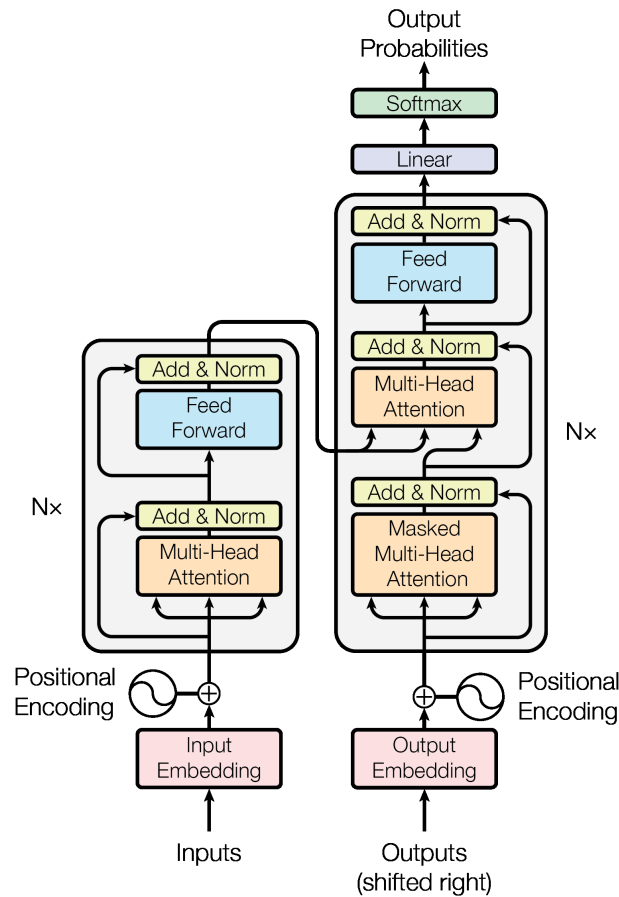


Figure 2.9: The Transformer model architecture. The encoder consists of N blocks on the left, while the decoder consists of N blocks on the right. (Source:[13])

residual connection. These components help to stabilize training by mitigating the issue of vanishing gradients.

Decoder: The decoder's objective is to generate an output sequence by employing the information that has been encoded by the encoder. The primary components of the decoder consist of:

1. **Multi-Head Self-Attention with Masking:** Like the encoder, the decoder's layers use multi-head self-attention. A masking strategy prevents tokens from training for subsequent places. Sequence generation activities that generate tokens from preceding tokens require this.
2. **Encoder-Decoder Attention:** Besides self-attention, the decoder uses encoder-decoder attention. This allows the decoder to generate tokens using encoder output. This helps produce contextually appropriate outputs based on input data.
3. **Position-wise Feed-Forward Networks:** The decoder layers have position-wise completely connected feed-forward networks. These networks use attention process data to understand complex relationships and make position-specific non-linear changes.

4. Layer Normalization and Residual Connections: Here, the decoder normalizes and links each sub-layer with a residual connection to ensure constant training.

Chapter 3

Related Work

This section provides an in-depth analysis of fundamental research in the fields of image captioning and neural machine translation. We aim to get a thorough comprehension of the progress made in image captioning, cross-lingual image captioning, and multilingual translation by examining critical research in this domain. The provided analysis serves as a meticulous examination of current research methods and frameworks, establishing the groundwork for subsequent chapters in this thesis

3.1 Image Captioning (IC)

Image captioning entails the automated generation of descriptive narratives for images, acting as a bridge between visual content and linguistic representation. The objective of this section is to examine significant developments and various approaches that have shaped the field of image captioning. We will examine how this area has developed over time, highlighting the most significant approaches and strategies that have emerged in response to its growing importance. In this section, the research is divided into two components. First, we investigate frameworks that follow conventional unilingual approaches. Next, we examine the field of cross-lingual and multilingual approaches.

3.1.1 Neural Image Captioning

In recent years, several methodologies have emerged with the goal of generating descriptive captions for images. Many of these techniques leverage RNNs, drawing inspiration from the successful application of sequence-to-sequence training of machine translation. The encoder-decoder framework, widely employed in machine translation, proves highly suitable for image caption generation as it effectively "translates" an image into coherent text. The initial foray pioneered by Kiros et.al. (2014a) [47] utilizes a feedforward neural network for predicting the next word based on the image and preceding word, while Mao et.al. (2014) [59] later replaced the feed-forward neural language model with a recurrent neural language model, employing a similar generation strategy. Another work by Kiros et.al. [48] suggests constructing a joint multimodal embedding space using a

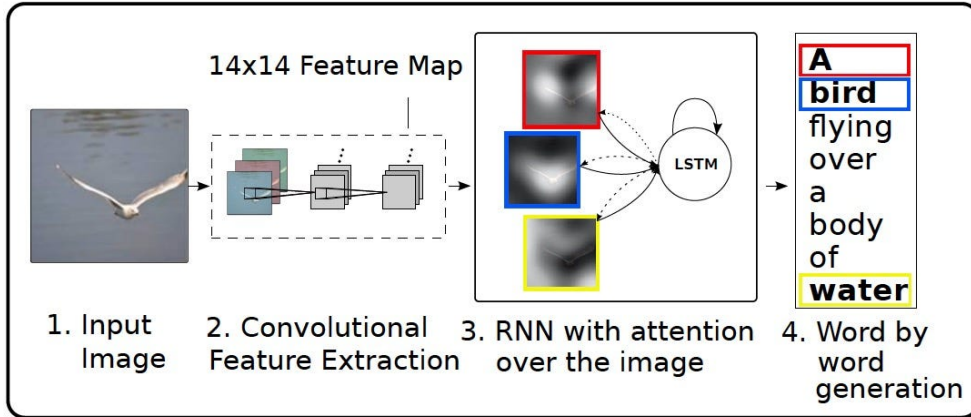


Figure 3.1: Approach Overview: In step (2), lower convolutional layers capture image features. Step (3), a feature is sampled and input to an LSTM to generate the corresponding word. Step 3 is iteratively repeated K times to produce a K -words caption. (Source: [84])

potent computer vision model and an LSTM for text encoding.

Diverging from the initial methods, and drawing inspiration from the recent success of end-to-end training in statistical machine translation [49], where direct maximization of translation probability achieves state-of-the-art results, Vinyals et.al. [78] adopt a similar approach for image captioning. Instead of utilizing an encoder recurrent neural network (RNN), they opt for a deep convolutional neural network (CNN) as the image encoder. Pre-trained on an image classification task, the last hidden layer of the CNN serves as input to the RNN decoder responsible for generating sentences. This model, known as the Neural Image Caption (NIC) [78], exhibits a positive performance correlation with the quantity of available training samples. Notable improvements are also observed on Flickr30k (56 to 66), SBU (19 to 28), and the recently released COCO dataset (BLEU-4 of 27.7). However, a limitation arises as the generated captions often focus on specific aspects of the image due to the model's singular presentation of the image at the beginning. In contrast to the approaches of Kiros et.al. (2014a) [47] and Mao et.al. (2014) [59], who incorporate the image at every time step, Vinyals et.al. (2014) [78] present the image solely at the beginning, leading to an inefficiency in leveraging the complete image representation to influence the formation of each word.

To overcome this limitation of NIC approach [78], incorporating an attention mechanism [76] has proven to be beneficial. Xu et al [84] describes an approach to caption generation with attention mechanism. The Attention-based encoder-decoder framework plays a crucial role by assigning importance to pertinent regions of the input image within the encoder network for generating each word in the decoder network. The central focus of the thesis work revolves around this approach, wherein we extend this idea to facilitate multilingual image captioning. Following that, I will provide a more detailed and thorough explanation of this paper. As shown in 3.1, the primary goal of the Image Captioning task is to construct a caption y encoded as a series of 1-of- K words, given an input image I [84],

$$y = y_1, y_2, y_3, \dots, y_c, y_i \in R^K \quad (3.1)$$

where K is the size of the vocabulary and C is the maximum sequence length.

Encoder Block (Feature maps from convolution layers): In this instance, a Convolutional Neural Model is used to extract features from the input image and represent them in a latent space, ensuring efficient encoding. Unlike previous studies that mostly rely on the flattened fully connected representation of the CNN, this work incorporates information extracted from the lower convolution layers. This decision is taken in order to maintain the alignment between the characteristics and the two-dimensional image. Moreover, this method allows the decoder network to focus on particular areas of an image by selecting a subset from the complete set of feature vectors. The output from the lower layers of the CNN is mainly in the format of kkD , where k denotes the size of the feature maps and D represents the number of convolutional filters. Next, we will transform the feature tensors into a shape of k^2D . For the sake of simplicity, let's suppose that k^2 is equal to L . Subsequently, the size of the obtained feature map can be represented as LD . The convolutional feature extractor generates L vectors, each with a dimension of D , which represent different parts of the image I ,

$$a = a_1, a_2, a_3, \dots, a_L, \text{ where } a_i \in R^D \quad (3.2)$$

where a as the final output from the Encoder Network.

Decoder Block (Generator Network-LSTM): The Decoder LSTM network functions as a generative network, producing one word at a time. It does so by taking into account previous hidden state, the words generated so far, and the encoder feature vectors. The vector $\hat{z} \in R^D$ represents the context vector, which captures the visual information linked to a certain input point. The authors introduce a method ϕ that calculates the value of z_t based on the annotation vectors a , where i ranges from 1 to L , corresponding to the characteristics retrieved from various image locations. The mechanism assigns a positive weight, denoted as i , to each location i . This weight can be interpreted in two ways: as the probability that location i is the correct place to focus on such that the text decoder can generate the correct next word, or as the relative significance of location i in combining the a_i 's together. Bahdanau et.al. (2014) [18] proposed a soft [76] variant of this attention technique. The weight i , of each annotation vector a_i is determined by an attention model called f_{att} . This model utilizes a multilayer perceptron that is conditioned on the prior hidden state h_{t-1} . The hidden state undergoes variation as the output RNN progresses in its output sequence. The network's next focus point is influenced by the sequence of words that have already been generated.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \quad (3.3)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})} \quad (3.4)$$

After calculating the weights, which add up to one, using the softmax function, the context vector \hat{z}_t is computed as follows:

$$\hat{z}_t = \phi(a_i, \alpha_i) \quad (3.5)$$

is a function that, when given a set of annotation vectors and their respective weights, returns a single vector.

Another paper by Biswas et.al. [20] describes an image captioning architecture incorporating a top-down attention mechanism. Stefanini et.al. [71] review numerous captioning

methods, datasets, evaluation measures, and visual encoding and text creation training strategies. The authors quantitatively compare numerous methods to discover the most influential architectural and training approaches.

Several datasets used are artificial, limited in size, or demonstrate bias towards specific topics, genres, or styles. An effective approach to tackle this problem is to employ data augmentation techniques, which can significantly expand the amount of training data that is accessible. Anagnostopoulou et.al. [16] proposed a technique for incorporating human feedback into the training procedure. This method allows for the creation of descriptions for fresh images by first training an image captioning model using the MS COCO dataset. Secondly, the users offer feedback on photographs in conjunction with their related subtitles. The feedback is subsequently employed to generate a supplementary training dataset, which is progressively integrated into the model's updates. In order to address problems such as catastrophic forgetting, sparse memory is utilized, and the effectiveness of feedback is further improved through the implementation of a data augmentation technique. Hartmann et.al. [36] implemented a system that optimizes the effectiveness of human feedback by utilizing data augmentation. The system consists of three primary components: feedback collecting, data augmentation, and model update. The system initially undergoes training using the MS COCO dataset and is then refined based on feedback from end-users, with a specific emphasis on sustaining user engagement. Collecting user input requires getting feedback of different complexities in order to achieve a balance between depth and user involvement, which in turn requires a well-designed user interface. Data augmentation involves utilizing feedback to create a more extensive collection of training samples, adopting various methodologies such as caption-based, image-based, or a combination of both. The enriched data is subsequently utilized to efficiently update the model parameters, resulting in improved performance. These studies provide comprehensive methods for data augmentation to increase the size of the training dataset in situations where there is a scarcity of data.

3.1.2 Cross-Lingual strategies in Image Captioning

In recent times, there has been a notable surge in scholarly investigations pertaining to the field of image caption generation. The majority of these studies have predominantly focused in the English language, primarily because to the extensive availability of dataset. However, the potential of image captioning should not be limited to a single language. Hence, cross-lingual image captioning holds significant importance for a substantial portion of those who do not speak English as their primary language.

Early research in this field tackled the problem by gathering large collections of image-caption pairs in the desired language. A notable example is the "YJ Captions 26k Dataset" [61], which is a Japanese version of the MS COCO [55] dataset. In this dataset, captions were generated by human annotators using Yahoo! Crowdsourcing services in Japan. Nevertheless, this approach is expensive and impractical, as it depends on human annotators and is time-consuming process. To address this, researchers are shifting towards learning from machine-translated text, making the process more scalable and cost-effective. Chen et.al. [24] were the first to attempt the elimination of the necessity for data annotation in every language, pioneering annotation-free cross-lingual image captioning evaluation. They introduced three metrics to evaluate the semantic coherence of captions without annotations. Their experiments revealed that machine-translated sentences, while occasionally grammatically correct, may lack fluency. For example,

English sentence: "A couple sits on the grass by the river with a baby and a dog"

German sentence: “Mit einem Hund und einem Baby sitzt im Gras an einem Fluss ein Paar”

This example demonstrates that although the keywords are translated correctly, the improper conjunction in the translated sentence diminishes its fluency. The problem of fluency becomes more noticeable as the length of the caption increases. The lack of fluency presents a difficulty in acquiring cross-lingual captions from machine-translated texts. Lan et.al. [52] proposed a fluency-guided learning system to improve the fluency and guarantee the grammatical correctness of generated captions. Unlike methods that rely on human curation, the objective is to completely train a cross-lingual captioning model using machine-translated sentences. In this research, the authors introduce a fluency-guided learning paradigm to tackle the fluency problems in translated sentences. This system comprises a module that automatically calculates the fluency of sentences and another module that employs these estimated fluency scores to efficiently train an image captioning model for the desired language. This technique enhances the fluency and relevance of generated captions in Chinese, without the need for manually written sentences in the target language. The findings suggest that less than 30% of the translated sentences satisfy the standards for fluency and need more improvement.

Deep learning for image captioning achieves impressive results but encounters a bottleneck due to the necessity for large annotated datasets, often scarce, expensive, and slow to acquire. In such situations, the use of unsupervised or semi-supervised techniques that can produce captions from unpaired data or leverage annotations from different domains or languages is extremely beneficial. A useful strategy for handling unlabeled data involves the application of transfer learning techniques [79], which have demonstrated notable benefits. Miyazaki et.al. [61] investigated the application of transfer learning in the field of cross-lingual image captioning. They pre-trained a model for English image captions and retained only one crucial layer, the one closest to the vision system, after removing all other trained layers. This layer transferred knowledge to Japanese by appending an untrained Japanese generation model to the English model. The bilingual model outperformed the monolingual one.

For another example, the authors Biswas et.al. [20] investigate a method for image captioning in German using transfer learning techniques. They proposed four methods, two baseline and two advanced, using the MS COCO dataset [55] (English captions) and the Multi30K dataset [29] (manually translated to German). Baseline 1 was trained on the translated MS COCO dataset, while Baseline 2 used a small Multi30K dataset. Advanced methods pre-trained on the translated MS COCO dataset and fine-tuned on German Multi30K were also explored. The first advanced method learned initial mapping from the translated corpus and then fine-tuned on Multi30K. The second advanced method employed an attention mechanism. Comparing the models, the one with the attention mechanism yielded promising results. Biswas et.al. [20] explore German image captioning through transfer learning, proposing and experimenting with four methods (two baseline and two advanced) using MS COCO and Multi30K datasets. The baselines involve training on translated MS COCO and a small Multi30K dataset, while the advanced methods are pre-trained on translated MS COCO and fine-tuned on the German Multi30K dataset. One advanced method learns an initial mapping from images to German, fine-tuned on Multi30K, while the other employs an attention mechanism with object-specific localized maps. The model incorporating the attention mechanism showed promising results in the comparison.

In certain investigations, the necessity for image-caption paired data in the target language is circumvented, leading to the concept of unpaired image captioning. This

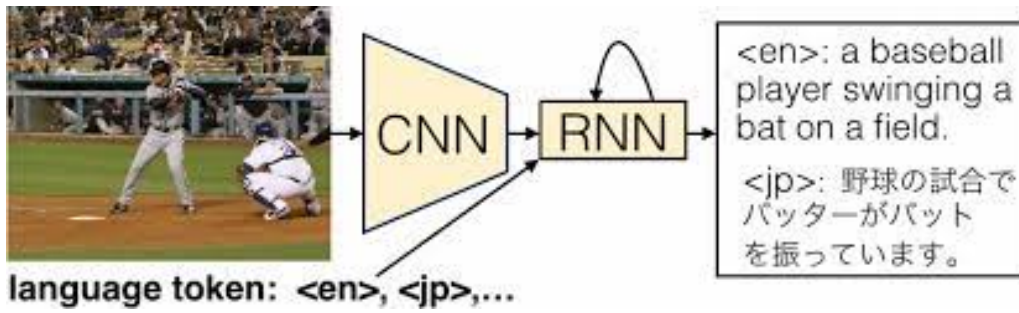


Figure 3.2: A unified model for multilingual captioning, employing artificial tokens to facilitate language switching (Source: [75])

method use a single encoder-decoder model to extract visual characteristics and produce captions in a pivot language, such as English. subsequently, a secondary encoder-decoder model aligns the pivot language caption with the target language caption, such as German. However, this technique has certain constraints: (i) Image captioning and machine translation are separate tasks that require different models and datasets; (ii) inaccuracies made by the image captioning model are propagated to the machine translation model. Gu et.al. [34] proposed a way to overcome restrictions by reducing differences between models and training them together to enhance interaction and learning. Their architecture comprises three models: an image captioning model to generate captions in the pivot language, a neural machine translation model to translate the caption into the target language, and a target language auto-encoder that directs the decoder to produce caption-like sentence. This approach produces captions in the target language that are reasonably adequate. Gao et.al. [31] proposed a two-phase approach. In the first phase, they use cross-lingual auto-encoding to train the mapping of a scene graph from the source language to the target language and then decode the sentence. In the second phase, they use cross-modal unsupervised feature mapping to learn how to map scene graph features from the image to the language modality. While both methods [[34], [31]] demonstrate potential for a multilingual captioning system, limitations arise, including a reduction in system quality as the number of languages increases, the necessity for re-training when adding a new languages, and an increase in the vocabulary that is shared between several languages.

To design a system generally accepted in industry, the adoption of a new language must be supported easily, while sustaining the system's precision and conserving resources and time. Tsutsui et.al. [75] proposed an approach that streamlines the addition of a new language to the system through the utilization of artificial tokens, enabling the integration of modules into a unified model. The idea is to add an artificial token at the beginning of each sentence, thereby governing the language of the caption. During the training phase, this token sent is the language of the ground-truth caption (e.g., <en> for English or <jp> for Japanese), and during testing, it tells the module to construct a sentence in the stated language. A part of the thesis work we utilizes the ideas from the aforementioned study, justifying a closer examination. The methodology utilizes a CNN (ResNet50 [37], pretrained on ImageNet [51]) to extract image features, subsequently used by an RNN (LSTM [39] with 512 hidden units) to generate captions. This approach aligns with previous methodologies [78]. Formally, the aim to minimize the negative logarithm of the likelihood of the caption given an image. A unique token, S_0 , signifies the sentence's start and the captioning language. In monolingual models, <sos> indicates

sentence initiation, while multilingual models use <en> or <jp> for English or Japanese (as shown in 3.2), respectively. The system, evaluated using the YJ Captions 26k Dataset [61], showed promising results in generating semantically meaningful captions. The experiments revealed the model’s proficiency in transitioning between languages, even dissimilar ones like English and Japanese, within a single neural model.

Another method for tackling cross-lingual image captioning entails integrating attention mechanisms. The study conducted by Wu et.al. [83] presents a two-part approach. Initially, the process of pre-training takes place on a model that includes an image encoder and an English decoder, specifically designed for generating captions in the English language. Furthermore, a German caption model is created, which consists of an encoder for English captions, a decoder for German captions, and a cycle consistency constraint. The image encoder generates English captions, which are subsequently used by another English encoder to produce a German caption. Cycle consistency is maintained through the utilization of three attention mechanisms: cross-attention between English decoder with respect to image regions, cross-attention between low-resource language decoder & feature maps from image encoder, and attention between low-resource language decoder conditioned on English words. The experimental results indicate that this strategy enhances the alignment between words and images and surpasses other methods based on common assessment criteria.

In conclusion, cross-lingual image captioning has witnessed a remarkable evolution over time. We’ve moved away from the labor-intensive process of manually annotating image-caption pairs in the target language to more efficient unpaired methods employing pivot languages as intermediaries. The integration of transfer learning has proven particularly valuable, bridging the gap when labeled data is scarce or extensive amounts of unlabeled data are involved. This advancement has substantially improved system performance, rendering it highly applicable for industrial use. Looking ahead, the promising realm of zero-shot learning holds potential for further enhancements in this evolving field.

3.2 Few-shot Image Captioning

Few-shot learning represents a paradigm-shifting approach in the field of machine learning, offering a solution to the conventional limitations associated with traditional supervised learning methods. Unlike traditional models that rely on explicit examples during training, few-shot learning empowers machines to generalize and make predictions with only a few training samples. This is particularly relevant in scenarios where acquiring large labeled training data for all possible languages is impractical or cost-prohibitive. This concept mirrors the human ability to transfer knowledge from known categories to new, unseen ones. Few-shot learning thus opens the door to more flexible and adaptive machine learning systems, capable of handling a broader range of tasks and domains without the need for large labelled data.

With the proposal of an ensemble-based self-distillation method, Chen et.al [25] present a novel approach to the few-shot image captioning field. By using unpaired images and captions, this technique makes it easier to train image captioning models and increases the model’s adaptability to a variety of data sources. The ensemble consists of several basic models that are trained using different data samples in each iteration to increase the resilience of the model. And use the ensemble to generate numerous pseudo captions, each given a weight based on the degree of trust in it, in order to efficiently learn from unpaired images. Furthermore, they offer a simple yet powerful pseudo feature creation

method using Gradient Descent [17] for learning from unpaired captions. The pseudo captions and generated pseudo features of the ensemble together help to train base models in further iterations, showcasing the adaptability and efficiency of the suggested method. In paper, [27] examines the difficulties associated with few-shot learning in the context of two multi-modal tasks: answering visual questions (VQA) and captioning images. Here, the authors introduce Fast Parameter Adaptation for Image-Text Modeling (FPAIT), an exciting method intended to simultaneously understand text and image data with sparse examples. FPAIT provides useful benefits in two main areas. In the first place, it demonstrates quick learning speed by obtaining suitable initial parameters for the joint image-text learner from a variety of tasks. With a minimal number of gradient steps, FPAIT effectively adapts to a novel assignment and achieves impressive results. Furthermore, FPAIT exhibits robustness against the constraints brought up by few-shot circumstances.

Contrastive pre-training has emerged as a potent strategy in achieving the goal of few-shot and zero-shot learning, bringing about a paradigm shift in the field of multi-modal research. Particularly in the intersection of vision and language, models such as CLIP (Radford et.al., 2021 [64]) and ALIGN (Jia et.al., 2021 [40]) have played a pioneering role by acquiring a shared multi-modal embedding space from extensive and noisy collections of image-text pairs. For instance, CLIP undergoes training on a dataset containing 400 million image-sentence pairs sourced from the web, leading to remarkable performance on tasks like image classification and vision-text retrieval. By adeptly crafting prompts, it becomes conceivable to improve the detection of objects that were not seen during training. Applications based on CLIP have demonstrated their proficiency in solving zero-shot problems across diverse and novel scenarios. It is noteworthy that zero-shot prompt engineering has also been applied to more advanced tasks, including Visual Question Answering (VQA), although its performance has not yet reached the levels achieved by supervised methods. Additionally, CLIP enhances text-driven image manipulation through the utilization of Generative Adversarial Networks (GANs) [32] or other generative models.

Furthermore, the few-shot learning method demonstrates remarkable flexibility. While the word-to-word metrics may be lower, the captions generated exhibit a strong semantic alignment with the image and convey real-world information, surpassing the restrictions typically associated with captions from human annotators in datasets used by supervised captioning methods.

3.3 Neural Machine Translation (NMT)

Statistical Machine Translation (SMT) [49] and phrase-based systems [50] are two traditional methods that are replaced by Neural Machine Translation (NMT), a major advance in machine translation. Driven by the growing demand for smooth communication across various languages in our globally interconnected society, NMT leverages deep learning to improve translation accuracy and fluency, leading to a revolution in the field of translation. Historically, SMT [49] held sway in machine translation, relying on statistical models to comprehend the dynamics between source and target languages through extensive parallel corpora. Despite notable achievements, SMT struggled with intricate connections and contextual nuances. To address these limitations, researchers proposed Phrase-Based Translation Systems [50], emphasizing the translation of phrases instead of individual words. Despite enhancements, these systems faced challenges in coherence and contextual understanding, especially with idiomatic expressions and com-

plex sentence structures. NMT's emergence marked a paradigm shift, leveraging deep learning techniques like RNNs and attention mechanisms. Unlike its predecessors, NMT designs do not use manual designs and rule-based systems, instead extracting knowledge directly from data. This shift enables NMT to capture intricate language patterns, resulting in more contextually aware and fluent translations. NMT boasts significant advantages over previous methods. Its capacity to learn complex relationships between words and phrases yields translations that are not only accurate but also contextually relevant. Additionally, NMT's superior generalization across diverse language pairs minimizes the need for language-specific customization. However, challenges persist, including addressing low-resource languages, mitigating biases in training data, and countering adversarial attacks, prompting ongoing research in these domains.

The practical implementation of machine translation has predominantly concentrated on certain language pairs due to the inherent challenges associated with developing a comprehensive system capable of consistently translating across multiple languages. The authors in [30] made an initial endeavor by modifying an attention-based encoder-decoder strategy to enable multilingual neural machine translation (NMT). This was achieved by incorporating independent decoders and attention mechanisms for each target language. Luong et.al. [56] explore the integration of multilingual training within a multitask learning framework. The existing model, an encoder-decoder network, lacks an attention mechanism. To harness multilingual data effectively, the researchers augment their model by introducing multiple encoders and decoders, each specifically tailored to support a particular source and target language. Unfortunately, the computational expense associated with working on large datasets and models renders the outlined systems impractical for both translation inference and training. Furthermore, conventional Neural Machine Translation (NMT) systems exhibit reduced reliability when handling input sentences containing uncommon terms, adversely impacting both accuracy and processing speed. Wu et.al. [82] addressed existing challenges by introducing Google's Neural Machine Translation system (GNMT). This paper provides a detailed account of the GNMT system's implementation, shedding light on crucial aspects such as model accuracy, robustness, and speed. The model comprises a deep LSTM network with eight encoder and decoder layers, incorporating residual and attention connections from the decoder to the encoder. The authors successfully enhanced the machine translation system, enabling effective performance on real data, accommodating large datasets, expediting translation inference, and improving translation quality and inference speed through adept handling of open vocabulary.

The described system is tailored for a unidirectional, language-specific model, designed for a single language pair. In a related development, Johnson et.al. [43] proposed a straightforward and effective method to handle multiple languages using a single model without necessitating significant alterations to the fundamental NMT framework. Their approach presented a simple and effective strategy⁰ for managing both high-resource and low-resource languages within a single model, preserving the fundamental GNMT architecture. The only modification involves adding an artificial token to the input sequence indicating the target language (e.g., for English to Spanish translation: <2es> Hello, how are you? -> Hola, ¿cómo estás?). This work highlights the potential of zero-shot translation, a successful demonstration of transfer learning in machine translation without additional steps. Despite numerous advantages, the system faced a limitation in incorporating new languages, requiring a complete system retraining, which is time-consuming and costly due to system expansion with additional languages.

Escolano et.al. [30] introduced a modular approach that enables the addition of languages without requiring the retraining of the entire system. This approach connects a shared

encoder-decoder model with a language-specific model, facilitating lifelong learning and the creation of a modular multilingual machine translation system. The system comprises two steps: initial joint training with language-specific encoder-decoder models, followed by the addition of new languages through the training of a new module connected to the existing ones. Upon experimentation, the authors identified three beneficial settings for the model: (i) a initial pretraining step on all combinations of English, German, French, and Spanish; (ii) incrementally introducing new languages, the authors tested with Russian \leftrightarrow English languages in a bi-directional setup; and (iii) In the final step, they perform Zero-Shot translation of all the languages on which the model was trained-on in set (i) & (ii). However, the system's cost is a notable concern, as training separate modules for each language demands significant memory and performance resources, and the effectiveness of zero-shot translation is not particularly impressive.

3.4 Summary

In summary, our exploration of zero-shot and cross-lingual image captioning underscores the transformative potential of these cutting-edge technologies. Cross-lingual image captioning, despite challenges in cultural nuances and translation accuracy, signifies a significant leap towards a globally interconnected digital landscape, fostering understanding across languages and cultures. Meanwhile, zero-shot image captioning represents a breakthrough in natural language processing and computer vision, enabling models to generate insightful descriptions for previously unseen images. Ongoing developments in zero-shot image captioning hold promise for improved accessibility to visual content and dynamic human-computer collaboration. The literature review reveals notable architectural advancements and performance enhancements in cross-lingual image captioning, emphasizing the dynamic nature of this evolving field.

Chapter 4

Implementation

This chapter explores the complexities of the methodology, implementation, and training strategies of our study. First we thoroughly examining the process of describing images using a single language model, delving into fundamental ideas, and seamlessly transitions into the challenges of multilingual captioning models. The discussion focuses on the integration of a pre-trained Neural Machine Translation (NMT) model to improve image captioning abilities, particularly in situations for low-resource languages. The chapter offers a thorough explanation of important choices, offering insights into the reasoning for selecting the NMT model as a crucial element in our efforts and how this helps us perform few-shot learning on low-resource languages. To conclude, the chapter outlines the intricacies of the training process, encompassing dataset preparation, the computational environment utilized, GPU setup, hyper-parameters, and the evaluation procedures employed in this study.

4.1 Methodology

The chapter initiates with an in-depth examination of the Single Model, 1-1 Model, and M2 architecture, framing our research around the [84] image captioning paradigm and the visual attention model [84], forming the foundational framework. It then explores various image captioning architectures, such as CNN-LSTM model, CNN-transformer model and CNN-pretrained transformer model. The overarching goal of this chapter is to furnish readers with a comprehensive understanding of the architectural evolution that shapes our distinctive image captioning approach. This comprehension lays the groundwork for the specific experiments detailed in the subsequent section.

4.1.1 Encoder Architecture

The encoder plays a pivotal role, serving as the cornerstone for extracting high-level features from input images. In the context of our experiments, the CNN encoder stands as a steadfast component, maintaining its architectural integrity across various models,

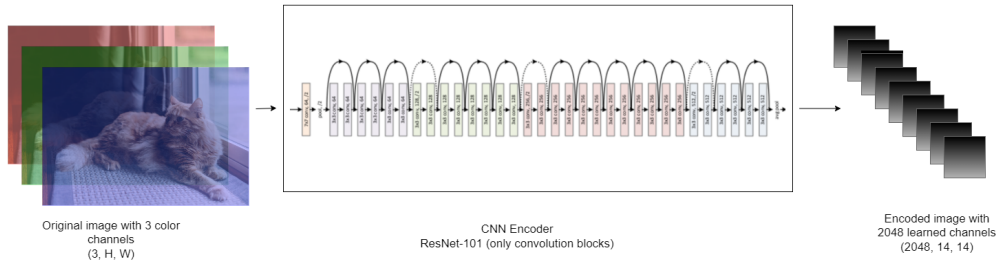


Figure 4.1: CNN Encoder - ResNet-101; Processes an image with 3 color channels, outputting feature maps through convolutional blocks

while we introduce variations in the decoder to explore diverse approaches to generating descriptive captions. Throughout our experiments, the CNN encoder remained constant, ensuring a consistent baseline for comparison across different decoding strategies.

The role of the encoder is to transform the input image, composed of three color channels, into a compact representation with learned channels. This method efficiently captures the crucial data embedded in the original image. Instead of training encoders, we use pre-trained CNNs that have already shown expertise in representing images. Therefore, the chosen architecture is the 101-layered Residual Network (ResNet-101), pre-trained on the ImageNet classification task and readily available in PyTorch [12]. This is a choice for its architectural features and performance benefits, specifically its superior capability to handle deep networks.

ResNet-101 progressively generates smaller representations of the original image, with each subsequent representation incorporating more learned features and an increased number of channels. The final encoded output is a tensor with dimensions $2048 \times 14 \times 14$, indicating 2048 channels and a spatial size of 14×14 . While the paper [84] mentions the use of a VGGnet, it emphasizes the need for modifications, particularly the removal of the last linear layers associated with softmax activation used for classification. For ResNet-101, the last two layers (pooling and linear layers) are discarded, emphasizing the exclusive focus on image encoding rather than classification. To ensure versatility in handling images of variable sizes, an adaptive AVG pooling layer is introduced to resize the encoding to a fixed size. For potential fine-tuning of the encoder, a fine-tuning method is incorporated, enabling or disabling gradient calculations for the relevant parameters. Notably, only convolutional blocks 2 through 4 in the ResNet are fine-tuned, preserving the foundational knowledge learned in the first block, which is crucial for basic image processing tasks such as edge and line detection. This decision aligns with the principles of transfer learning, allowing for the option of fine-tuning to enhance performance.

4.1.2 CNN-LSTM Architecture

The decoder's function in image captioning is to iteratively generate a caption based on the encoded image, employing an RNN with a Long Short-Term Memory (LSTM) network. Operating as a generative network, the Decoder LSTM produces words one at a time, taking into account the prior hidden state, previously generated words, and encoder feature vectors.

In a conventional setting without attention, the encoded image could be averaged across

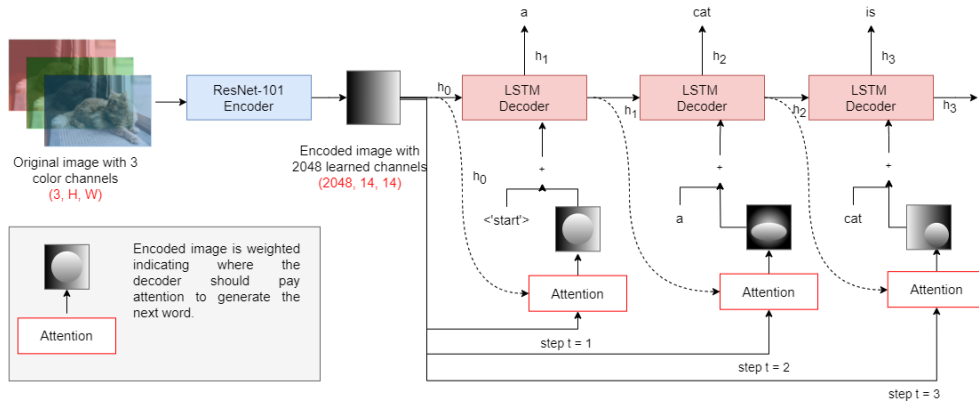


Figure 4.2: CNN- LSTM approach: Image features are captured at lower convolutional layers, sampled, and fed to LSTM for generating corresponding words. This process is repeated K times to produce a K-words caption

all pixels, and this average, possibly linearly transformed, would serve as the initial hidden state for the decoder, as suggested in the [78]. However, in the presence of attention, the decoder aims to focus on different regions of the image during various stages of caption generation. This is achieved by using a weighted average across all pixels, assigning higher weights to more important pixels. The resulting weighted image representation is then concatenated with the previously generated word at each step, guiding the generation of the next word. The work [84] examines two attention mechanisms, specifically hard attention and soft attention. However, in our study, we mainly utilize the soft attention mechanism [76].

The output of the Encoder, initially in dimensions $N, 14 * 14, 2048$, is flattened for convenience, avoiding the need for multiple tensor reshaping operations. To streamline the decoding process, images and captions are sorted by decreasing caption lengths, facilitating the processing of valid timesteps and excluding padding tokens. The manual iteration over each timestep is executed in a loop, as opposed to a continuous iteration. This manual iteration is necessary for incorporating the attention mechanism between each decoding step. The attention-network computes model weights and attention-weighted embeddings at each time-step. The paper [84] recommends passing the attention-weighted embedding vector through a sigmoid filter to add non-linearity & crop the gradients between a range of -1 to +1.

The resulting embedding vector is concatenated with the embedding of the previous word - "<start>" to signal to the decoder to start the caption generation. Finally, run the LSTM Cell to generate the output logit. A fully-connected layer then transforms this logit into scores for each word in the model's vocabulary. The scores are then converted into a normalized probability distribution which sums to 1. Additionally, the weights returned by the attention network at each time-step are stored for further analysis. The detailed decoding process outlined here illustrates the intricacies of incorporating attention mechanisms to enhance the image captioning model's ability to focus on relevant image regions during sequential word generation.

4.1.3 CNN-Transformer Architecture

Within the field of image captioning, the Transformer architecture has exhibited exceptional ability in capturing complex interconnections and contextual relationships within sequential data. In our approach we enhance the capabilities of the Transformer model during the decoding phase. We are conducting studies that involve both the basic Transformer model and a pre-trained version. More precisely, within the decoder of the pre-trained Transformer, we are currently investigating alterations by replacing the fully connected layer with 1x1 convolution layers.

Marian NMT

We are using pre-trained Marian NMT [44] models from Hugging-Face [9]. MarianNMT adopts an encoder-decoder architecture and was initially developed by Jörg Tiedemann. The models were pretrained mostly on the Open Parallel Corpus (OPUS) [73], which is a collection of translated texts from the web. Thus the models offer support for multiple language combinations. Lastly, the models adhere to a consistent naming convention – *Helsinki – NLP/opus – mt – src_language – target_language*".

Pre-trained MarianNMT transformer decoder

The output tensor from the CNN encoder, representing image features, is a 3D tensor of shape (14, 14, 2048). To leverage the strengths of the Transformer architecture, segment this tensor into 196 vectors, each encapsulating a 2048-dimensional feature. However, before feeding these vectors into the Transformer encoder, introduce a positional-encoding mechanism. Unlike sequences, where the order is inherent, images lack a natural order for the Transformer to recognize. Our positional-encoding strategy is spatial in nature, accounting for the 14x14 grid of pixels in each image feature. For each pixel at position (x, y), create a 1024-dimensional vector to represent its vertical position (x feature) and another 1024-dimensional vector for its horizontal position (y feature). Concatenating these two vectors results in a 2048-dimensional position feature vector, ensuring the Transformer captures the spatial relationships crucial for image understanding. The architecture is illustrated in Fig.4.3.

The decoder input of the Transformer also requires positional-encoding, following a conventional method. Post positional-encoding, the encoded image features are fed into the Transformer encoder, extracting a comprehensive representation of the image's content. During training, the entire target captions are provided to the decoder part of the Transformer, enabling predictions for all sequence positions simultaneously. This contrasts with traditional LSTM architectures, where only one next word is predicted at a time during training. The efficiency of the Transformer becomes particularly evident in its faster training pace compared to LSTM. In the caption generation phase, the process aligns with LSTM, predicting one word at a time based on the previously generated word during evaluation. This adaptation of the Transformer architecture for image captioning underscores its versatility and efficiency in handling complex sequential data.

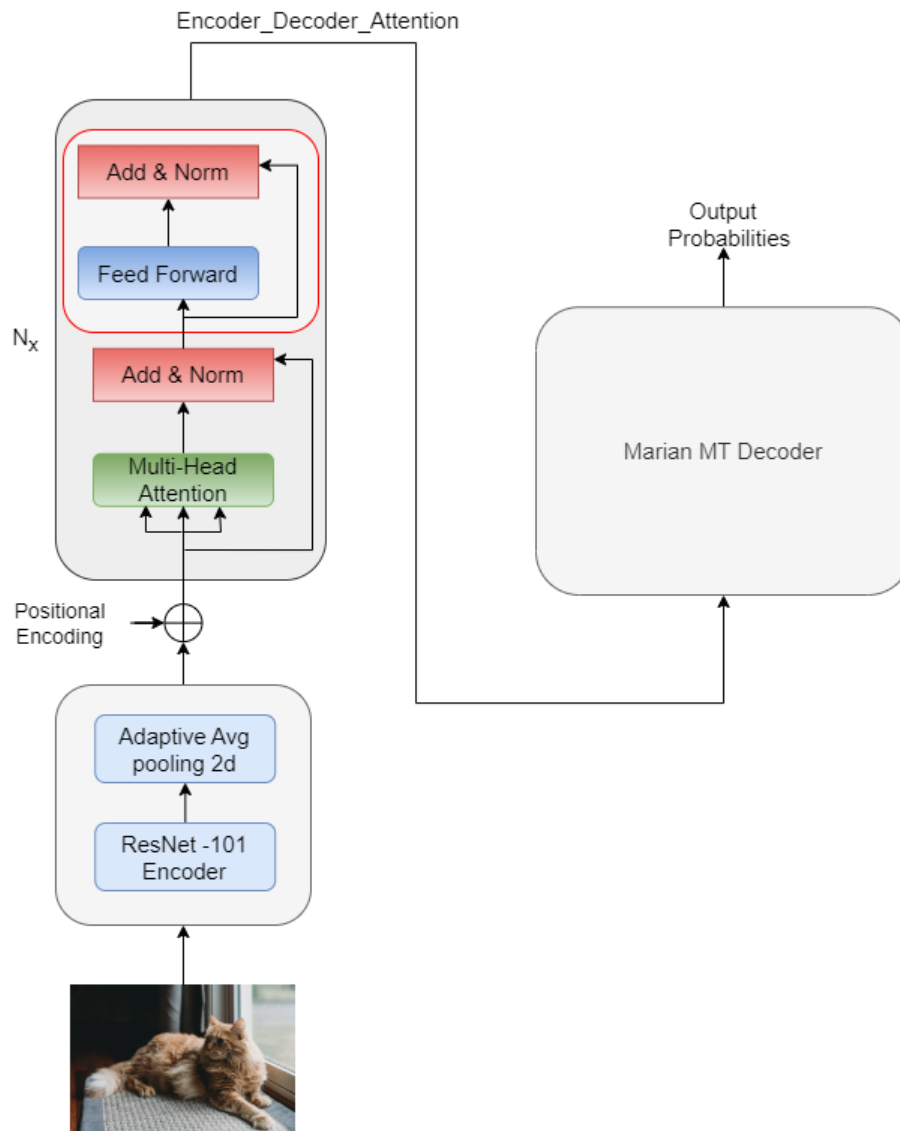


Figure 4.3: A high-level overview of an image captioning system which incorporates a ResNet-101 model as image Encoder and Marian NMT model as auto-regressive text Decoder

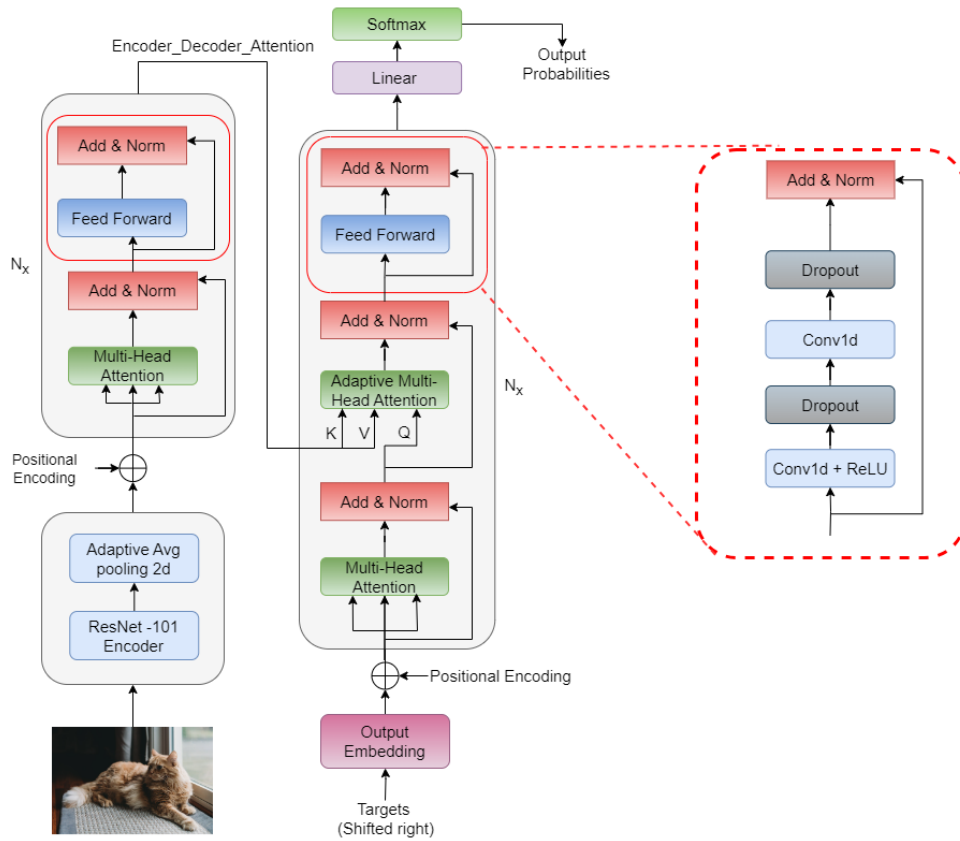


Figure 4.4: An expanded illustration of the modifications made to the Marian NMT auto-regressive text decoder. The section inside the dashed lines illustrate the addition of 1D convolution layers to extract information from the feature maps passed by the encoder

Pre-trained MarianNMT transformer decoder with 1x1 CNN-based cross attention layers

From the insights gained in 4.4, our implementation adheres to the fundamental transformer architecture with a notable modification. We replace the the full-connected layers with 1x1 convolution layers.

A 1x1 convolutional layer is often referred to as a point-wise convolution. A 1x1 convolutional layer with N filters operates on each pixel independently across channels and is mathematically equivalent to a fully connected layer with N neurons. A fully connected layer performs a weighted sum of its input neurons, where each input is multiplied by a corresponding weight, and the results are summed up. A 1x1 convolutional layer with N filters can be seen as performing a weighted sum of its input channels at each spatial location. The advantages of employing 1x1 convolutions include parameter sharing, preserving spatial hierarchy, computational efficiency, non-linearity, and adaptability to variable input sizes. Notably, parameter sharing exploits spatial locality, reducing the number of parameters compared to fully connected layers. The spatial hierarchy is maintained, allowing the recognition of local patterns and their combination into higher-level representations. Computational efficiency arises from optimized convolutional algorithms, contributing to faster training and inference times. The non-linear nature enhances the model's adaptability to intricate patterns, while the ability to handle variable input sizes is crucial in applications with diverse image dimensions.

4.2 Data Pre-processing

We customize the data preparation process according to the various methodologies explored in our study, specifically the single model, 1-1 model, and M2 architecture. We using Andrej Karpathy's predefined training, validation, and test splits [45] that were initially designed for the English MS-Coco dataset. Consequently, we redesign this format to support datasets in Italian, Spanish, and a mixed dataset that incorporates a shuffle of all three languages. Before being organized into Andrej Karpathy's specified splitting format, the raw captions undergo preprocessing and tokenization. This format includes essential details such as the image filepath, image id, image file name, sentences with tokens, raw captions, sentence id, and image id. The data is stored in a JSON format and serves as the input for the subsequent preprocessing step, along with the COCO dataset images folder as illustrated in Fig. 4.5. The necessary inputs for the model, include images, captions, and caption length.

1. **Images:** Since we are using a pre-trained ResNet-101 Encoder, it is crucial to pre-process the images to meet the requirements of the pre-trained Encoder. The pre-processing consists of two essential steps: firstly, ensuring that the pixel values are limited to the range of $[0, 1]$, and secondly, normalizing the images by using the mean and standard deviation obtained from the RGB channels of ImageNet images. In order to ensure uniformity, all MS-COCO images are resized to dimensions of 256×256 . Consequently, the model requires the input images to be in the form of a Float tensor and necessitates their normalization using the mean and standard deviation. Here N is the batch size. In order to optimize the management of images during the process of training and validation, we save them in an HDF5 [6] file. Each split in the HDF5 file is stored as a tensor. It is important to mention that pixel values are kept within the range of $[0, 255]$. The decision to use HDF5 files is based

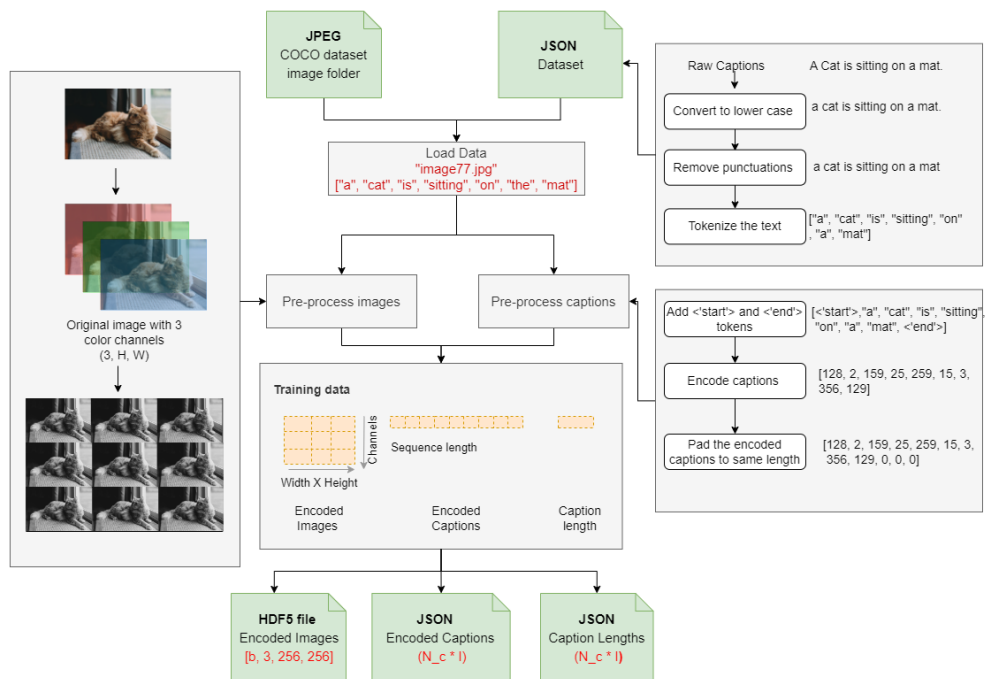


Figure 4.5: Left-most section: convert the RGB image into grey scale. Right-most section: Converts the raw captions into encoding. Mid section: pre-processed images and captions are serialized and stored in binary file format such as HDF5

on the practical constraint that the photos are too huge to fit in the computer’s memory. Therefore, we directly access and retrieve them from the disk during the training and validation processes. The steps for pre-processing images remains same for all three approaches.

2. **Captions:** The raw sentences are extracted from Andrej Karpathy’s split JSON file and then tokenized according to the specific model being employed. The NLTK [10] word tokenizer is used for the CNN-LSTM model, whereas the tokenizer defined in the NMT model provided by Hugging Face is utilized for the CNN encoder - Marian NMT decoder model.

The captions function as both the outputs and inputs for the decoder, as every word is utilized for generating the subsequent word. Nevertheless, in order to commence the process of generating captions, it is important to have a zeroth word, denoted as <start>, which is crucial for anticipating the first word. Likewise, when approaching the final word, the model is trained to predict the occurrence of <end> token. This capability is essential as it allows the Decoder to determine when to terminate the decoding operation during inference. An example of an input sequence looks like,

```
<start> a cat is sitting on a mat <end>
```

To provide a consistent length, captions need to be padded using <pad> tokens, as they are considered fixed size tensors. Furthermore, it is imperative to have a vocabulary file that serves as a comprehensive index, mapping each word in the corpus. This vocabulary file includes the <start>, <end>, and <pad> tokens. Therefore, the captions provided to the model should appear as an Integer tensor.

3. **Caption length:** As the captions are subject to padding, it is crucial to closely control the lengths of each caption. The length of the caption is determined by the sum of the actual length and the length of the <start> and <end> tokens. This strategy aims to optimize computation by eliminating <pad> tokens and processing a sequence only up to its length. Therefore, the model requires caption lengths to be provided as an Integer tensor. The caption length are then stored in the a JSON file.

4.3 Approaches

In this section, we elucidate the methodologies employed in our study, encompassing the Single Model, 1-1 Model, and M2 Model. Each approach is distinct in its design, data pre-preparation and training procedures, contributing to a comprehensive understanding of the diverse strategies applied in our investigation.

4.3.1 Single model

Establishing a multilingual captioning system [84] can be achieved by opting for a separate model for each individual language. However, this necessitates the creation of a model for each supported language, leading to scalability challenges. The initial approach involves employing multiple unidirectional models, where each model comprises an

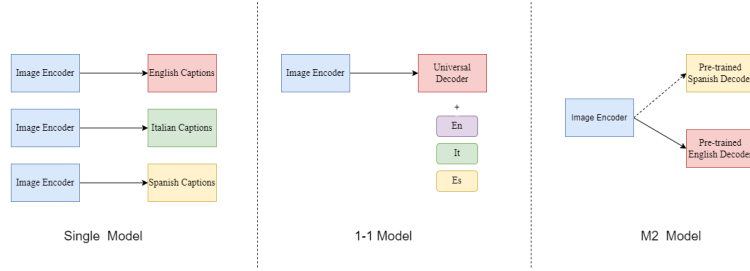


Figure 4.6: **Overview** of three distinct multilingual image captioning models designed for the languages English (En), Spanish (Es), and Italian(it). Left-most section: consists of a collection of individual models designed for 3 different directions. Mid section: a 1-1 model that shares all the parameters of the model . Reight-most section: the M2 approach primarily shares language-specific modules.

encoder for processing the input image and a decoder for generating captions in the respective target language. As illustrated in Fig. 4.6, for the Single model there are distinct image encoders paired with decoders for different languages, such as English, Italian, and Spanish. To implement this, we leverage the models mentioned in Section.4.1, and the data pre-processing steps remain constant and aligned with the steps provided in Section.4.2.

Moreover, these models demand a substantial amount of data in the targeted language, and availability of data for image-caption pairs in languages other than English is often constrained. Consequently, this approach becomes impractical, particularly as the number of languages increases, leading to an exponential growth in the number of required models. In practical image captioning applications, the need arises to effectively handle multiple languages. However, training separate models for each language is not a practical solution. Hence, there is a crucial necessity to develop a unified model capable of supporting and accommodating multiple languages efficiently.

4.3.2 1-1 model

The proposed approach suggests training a unified caption generator capable of producing meaningful captions in multiple languages. This method proves to be more practical than a single-model approach, as it efficiently reduces the number of models by sharing components among them. The 1-1 method, in particular, employs a single encoder and a single decoder to facilitate captioning in multiple languages. This approach stands out for its compactness, significantly reducing the number of parameters while concurrently enhancing the system's overall performance.

Notably, the implementation of this method requires no alteration to the conventional image captioning system. Instead, modifications are made to the data, leaving all other components of the system—encoder, decoder, attention mechanism, and vocabulary, as detailed in section.4.1 unchanged. This ensures a seamless integration of multilingual capabilities into the existing system without the need for substantial structural modifications. Data preparation follows the same procedures as outlined in Section:4.2, with minor adjustments made to the captions and the images. In order to effectively employ multilingual data within a unified system, we advocate for a straightforward modification to the input data.

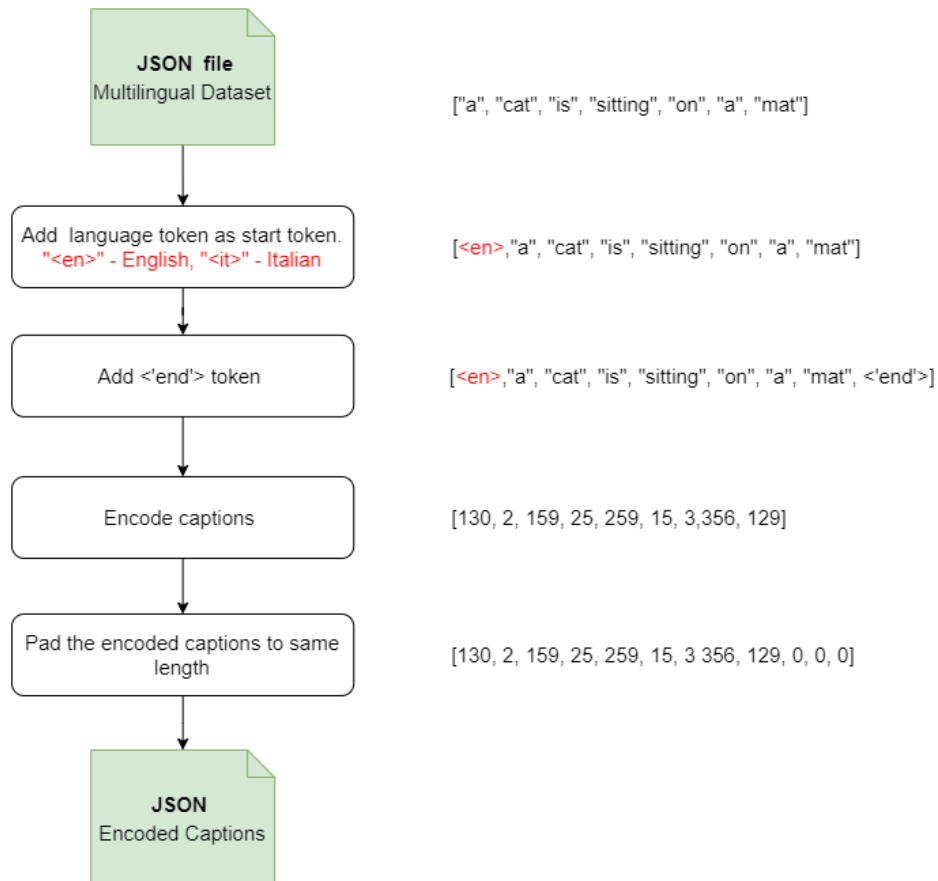


Figure 4.7: Data Pre-processing: inject language token at the beginning of the captions to instruct the model to generate caption in the target language

1. **Injecting language token in the beginning of the captions:** To indicate the desired language for the model to generate captions in, a language token is included at the start of the input sentence. To visually represent this concept, refer to Fig. 4.7, where the captions undergo tokenization with the language token injected as the initial token. Let's analyze the pair of sentences below:

```
<en> "a", "cat", "is", "sitting", "on", "a", "mat" <end>
<it> "un", "gatto", "è", "seduto", "su", "un", "tappeto" <end>
```

where, <en> represents English language token and <it> for Italian. Once the language token is incorporated into the input captions, we proceed to train the model using multilingual captioning data, which encompasses English, Italian and Spanish languages. During the inference providing the language token requests the model to generate captions in the target language.

2. **Languages token overlay on the image:** An experimental scenario was conducted wherein language tokens corresponding to English (en), Italian (it), and Spanish (es) were injected onto the images. The data pre-processing involves two main steps: a) the initial step, identical to the process outlined in Section.4.2, includes an additional JSON file storing the language tokens for the captions; b) the second step involves loading the files saved in step a, specifically the images stored in an HDF5 file, which are deserialized, reshaped, and then overlaid with language tokens, as illustrated in Fig. 4.8. Following the incorporation of language tokens into the input images, the model is trained with multilingual captioning data, encompassing multiple languages.

To enhance cross-modal grounding [87], we implemented a strategy to address a common issue where the model tends to generate captions based on high-frequency words, often overlooking the actual content of the image. As illustrated in fig. 4.9, the model accurately predicts identical captions for two separate photos, highlighting the superior influence of the language decoder. In Image 1, the caption "the cat is sitting on a mat" is accurately generated, but in Image 2, the model mistakenly outputs the same caption. Cross-modal grounding is essential for addressing these difficulties by improving the model's ability to understand information from many modalities, thus strengthening the model's robustness.

The main benefit of the model is its simplicity. As there are no alterations to the model architecture, expanding the model to accommodate more languages is a straightforward process. Additional data can be seamlessly incorporated by augmenting the dataset, potentially employing over- or under-sampling techniques to ensure a balanced representation of all languages. The introduction of a new token becomes the sole adjustment needed when transitioning to a different target language. The training procedure remains unchanged, with mini-batches for training being sampled from the combined mixed-language training data, mirroring the process in the single-language scenario. The deployment of such a multilingual model in production is also notably simplified, as it effectively reduces the overall number of required models when managing multiple languages.

However, the 1-1 model is not without its limitations and encounters capacity bottlenecks that are less than ideal. A capacity bottleneck arises when the model becomes constrained by the trade-off between the number of tasks introduced and captioning accuracy. In

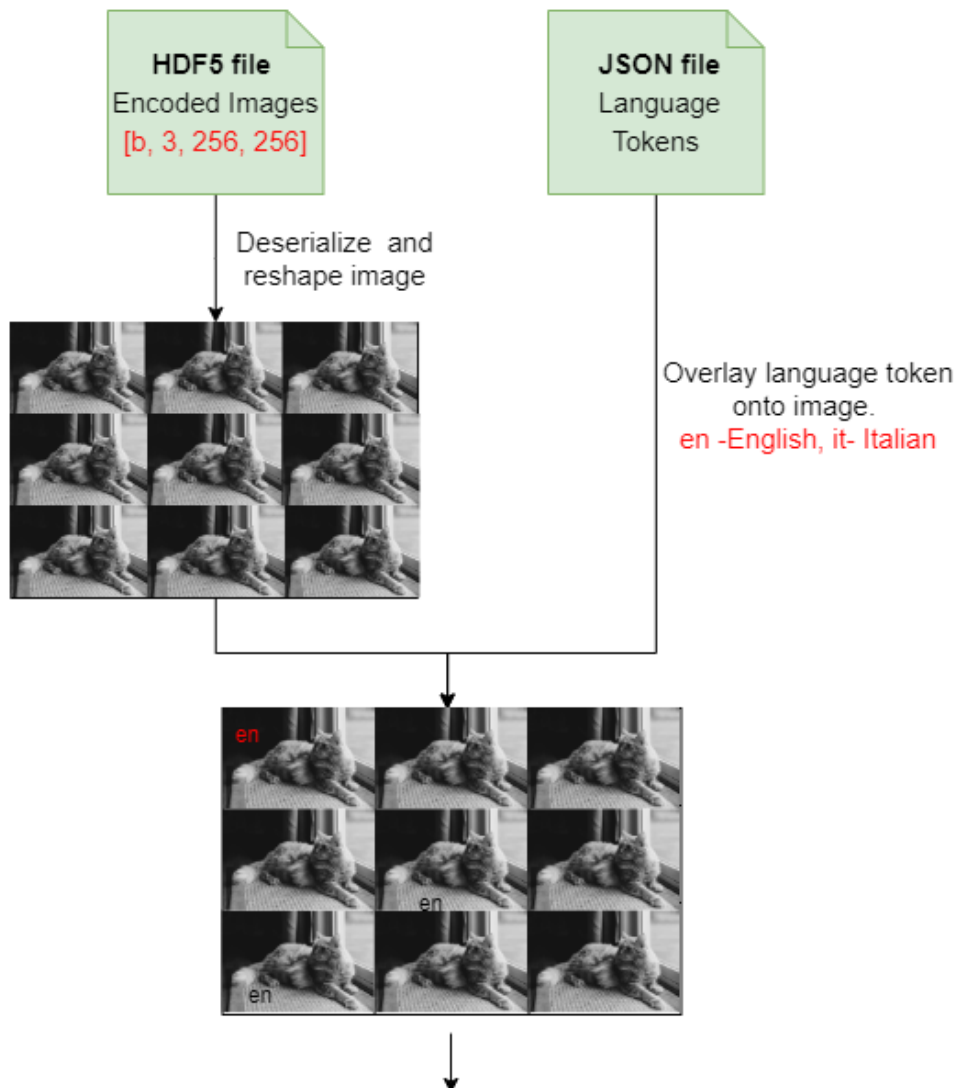


Figure 4.8: Data pre-processing: overlay language token on the image in three different positions with two different colors, to guide the model generate caption in target language

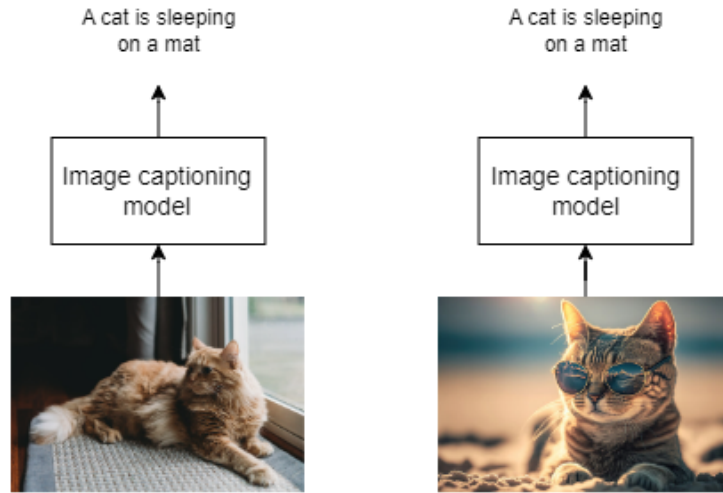


Figure 4.9: An example explaining cross-modal grounding. The left-most section: model generates a accurate captions extracting information from multiple modalities. The right-most section: the model generates inaccurate caption, which suggests that the model lacks the ability of extract information from both modalities

other words, performance may decrease when the number of image captioning directions is doubled. The 1-1 model, despite its benefits in multi-way training, exhibits lower maintainability, making it less appealing for industrial applications. Additionally, incorporating a new language into the system becomes a cumbersome task as the entire model necessitates retraining as one unified entity, demanding significant effort and time.

4.3.3 M2 architecture

The M2 method serves as the foundational framework for our thesis. This approach stands as a viable alternative to the 1-1 model, offering a solution that aligns with industrial requirements. This proposed method adopts an efficient architecture known as the modularized multilingual NMT model (M2). Unlike the 1-1 model, the M2 model selectively shares language-specific modules, specifically either the encoder or the decoder. The authors substantiate that the M2 model effectively addresses the limitations observed in the 1-1 model, leveraging the advantages of multi-way training without succumbing to the capacity bottleneck. Coupled with its modularized architecture, M2 facilitates a convenient and efficient modification of the model.

The construction of the M2 architecture in few-shot learning scenario involves two key steps.

1. At first, NMT models is a requirement, and these models are trained using parallel data. Illustrated in Fig. 4.10.a, our exemplar scenario employs a German Encoder trained with an English decoder, followed by the utilization of the same German encoder with a Spanish decoder. This approach ensures compatibility between both decoders, allowing them to be effectively employed with the same encoder. The incorporation of pre-trained NMT decoders from Hugging Face, as delineated

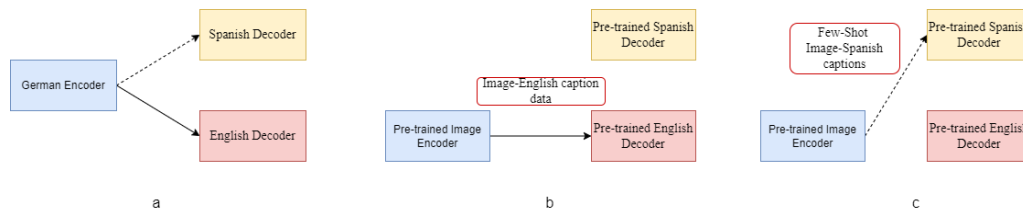


Figure 4.10: Left-most section: NMT model pre-tainting, an encoder-decoder architecture to train a German-English translation model and using the same German encoder to train a German-Spanish translation model. Mid section: Train a image captioning system, with image encoder and English NMT decoder. Left-most section: use the pre-trained image encoder and pre-trained NMT Spanish decoder to perform few-shot learning

in Section 4.1, forms a crucial component of our experimental setup for the M2 architecture.

2. After obtaining our pre-trained NMT decoder, as depicted in Fig. 4.10.b, the subsequent step involves the training of our image encoder using an image-captioning dataset. In this process, only the image encoder is trained, while the NMT decoder remains frozen. This strategic approach ensures that the image encoder captures the intrinsic characteristics of the images, preparing it for utilization with other language decoders for subsequent few-shot learning tasks.
3. Following the training of the image encoder, we proceed to perform few-shot learning using other language NMT decoders. Fig. 4.10.c illustrates the execution of few-shot learning with a Spanish NMT decoder. Leveraging the pre-trained image encoder and language decoder allows for the facilitation of cross-lingual transfer learning, particularly beneficial for low-resource languages. This approach enables effective few-shot learning, showcasing the versatility and adaptability of the model across diverse linguistic scenarios.

High-resource languages play a pivotal role in bolstering the capabilities of natural language processing models for low-resource languages. Leveraging pre-trained models from languages with abundant linguistic resources offers a multitude of advantages. These pre-trained models serve as a foundation, enabling faster convergence during training and improved performance with limited data. The availability of large datasets for high-resource languages also facilitates data augmentation for their low-resource counterparts, enriching training sets and enhancing model generalization. Techniques such as fine-tuning and transfer learning allow the adaptation of pre-trained models to the specific linguistic nuances of low-resource languages. Moreover, shared linguistic features across languages and the application of resource-efficient techniques contribute to more effective and accurate models for languages with fewer available resources. This collaborative synergy between high-resource and low-resource languages accelerates progress in natural language processing across diverse linguistic landscapes.

4.4 Training details

4.4.1 Loss Function

The loss function computes the overall loss by employing CrossEntropy to compare the predicted captions with the target captions, just utilizing the unprocessed scores obtained from the final layer of the decoding process. The author of [84] proposes the use of a second loss function called the attention alpha loss. The alpha loss aims to promote a fair allocation of attention between the decoder and encoder over the entire image, discouraging an excessive focus on any specific region. This modification seeks to optimize the attention system's efficiency in extracting information from the full image, hence reducing the likelihood of creating captions that include redundant words. Furthermore, we exclude the padded portions in a sequence when calculating the loss.

4.4.2 Beam Search

A straightforward approach is greedy search. The primary limitation of greedy search is that it is prone to yield suboptimal solutions. Due to the independent nature of each decision at every step, there is no regard for the cumulative effect on the whole sequence. The model might choose a word with a relatively high probability at a particular phase, but this decision may not be optimal for the overall coherence and accuracy of the sequence. For instance, in the context of caption generation, selecting the word with the highest probability at each stage may lead to a caption where the chosen words individually possess high probabilities, but the overall sequence may lack fluency, coherence, or even accuracy.

In order to overcome this limitation, different search strategies, such as beam search are frequently used. These methods employ a systematic approach that examines many options at each stage, taking into account a wider range of factors and enhancing the probability of discovering a solution that is globally optimal. In our experiments we utilize beam search. To better understand the procedure of beam search, refer to Fig.4.11. The process of beam search can be explained by the following steps:

1. First consider the top k candidates. (Example: an, a, the)
2. Generate k subsequent words corresponding to each of the initial words. (Example: an cat, a cat, a man)
3. Select the top k word combinations, ranked by their score. (Example: a cat, a man)
4. For each of the second k words, select the third word and then select the top combinations of words. (example: a cat is, a cat sits)
5. Implement this for every decode step.
6. Once the sequence comes to an end, indicated by the presence of the <end> token, select the caption with the highest overall score. (Example: a cat is sitting on a mat.)

4.4.3 Early stopping

We make use of the BLEU-4 score evaluation metric in order to assess the performance of the model on the validation set. This metric compares the generated captions with the

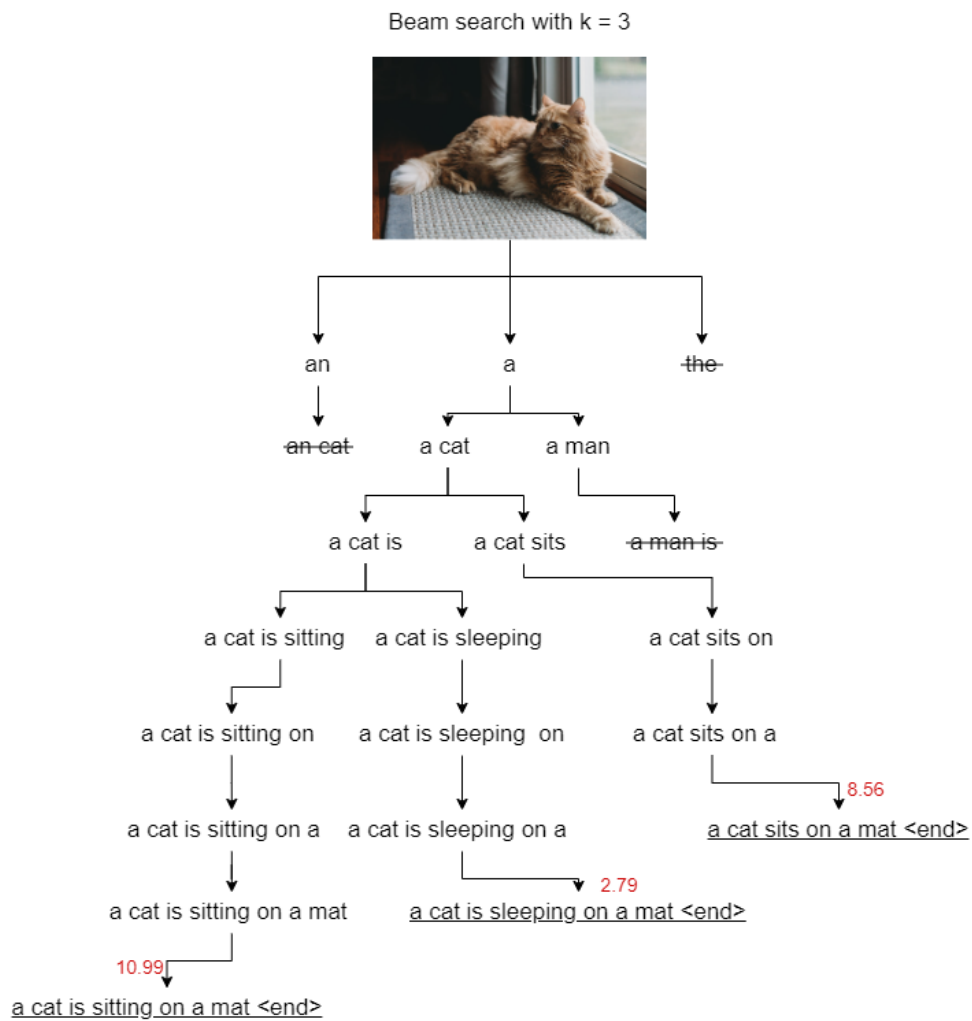


Figure 4.11: Illustration of Beam Search: A graphical depiction illustrating the consecutive stages of beam search. It demonstrates the multiple potential paths to generate more accurate and contextually relevant sequences

reference caption. Each generated caption is evaluated by comparing it to all available captions for the corresponding image, which serve as reference captions.

Consistent with the conclusions of the [84] study, it is important to highlight that the relationship between loss and BLEU score weakens beyond a particular threshold. Hence, the authors suggest stopping the training process at an early stage when the BLEU score starts to decline, regardless of the ongoing decrease in loss.

4.4.4 Training details

This report provides an in-depth exploration of the training complexities for three distinct models: the Single model, 1-1 model, and M2 model. Every model goes through customized training procedures, with a main focus on improving the decoder and, in certain cases, adjusting the encoder.

1. **Single model training:** We initiated the training procedure by focusing exclusively on improving the Decoder, while maintaining the Encoder frozen in the beginning. The training process was optimized by using a batch size of 128. The Adam [46] optimizer was employed, with a decoder learning rate set to $1e-4$. In addition, a gradient clipping [85] threshold of 5 was applied to reduce any potential problems connected to gradients throughout the training process.
2. **1-1 model training:** In the 1-1 model scenario, we experimented two distinct approaches: one where the language token was inject at the beginning of the captions and another where the language token was overlaid on the image. For both the approaches training began exclusively for the decoder, similar to the Single model approach. The same batch size of 128 and Adam optimizer with a decoder learning rate of $1e-4$ were employed. A gradient clipping threshold of 5 was maintained as a precautionary measure throughout the training process.
Subsequently, the Encoder was fine-tuned, specifically when language tokens are overlaid on an image. During the process of transfer learning, we took a careful approach, recognizing that the model we used was already trained on a different task. Therefore, a reduced learning rate of $1e-4$ was used for the purpose of fine-tuning the encoder. This adjustment accounts for the necessity of maintaining the characteristics obtained by the pre-trained model, while avoiding drastic modifications that might disrupt its learned representations. This ensure a careful modification of the image understanding component to align with the introduced language tokens.
3. **M2 model training:** For the M2 approach we again experiment with two distinct scenarios: Marian NMT decoder and modified Marian NMT decoder. And we trained the model to perform few-shot learning in two steps of training:
 - (a) The M2 model training commenced with a distinctive approach for the image encoder model. In this case, the Marian NMT decoder was initially kept frozen. But in case of modified Marian NMT architecture, training focused also on refining the decoder, specifically the added 1×1 convolution layer, while keeping the rest of the decoder frozen. A batch size of 128, Adam optimizer, and a encoder and decoder learning rate of $1e-4$ were employed. A gradient clipping threshold of 5 was implemented as a precautionary measure.
 - (b) After completing the training of the image encoder, we seamlessly transitioned to swap the English language decoder with the Spanish decoder. With the

Tools and language	Remarks
Python	open-source library support, building pipeline, multiprocessing, community support, wrapper
Numpy	Efficient computational operations on multi-dimensional arrays
Pandas	data manipulation, data analysis, data cleaning
PyTorch	tensor operations, GPU support
NLTK	BLEU score computation, word tokenizer
HuggingFace	MarianNMT models, MarianNMT tokenizers
A100, V100, RTX A6000 GPU	For the effective training and evaluation of models
Matplotlib	graph visualization
Wandb	to monitor and display model training in real time
Github	version control

Table 4.1: List of the technological components used in this study, supplemented by remarks highlighting their usage and importance

pre-trained image encoder and the MarianNMT Spanish decoder in place, we conducted training using a limited number of examples to facilitate effective few-shot learning.

4.4.5 Hyper-parameter tuning and Computational Setup

Hyperparameter tuning is crucial for optimizing the performance of image captioning models. Throughout the experimental phase, we methodically investigated different hyperparameters in order to enhance the overall effectiveness of the models. A range of learning rates around $1e-4$ was thoroughly tested to determine the most suitable values for both the decoder and encoder. Additionally, the beam size during the decoding process has been identified as a crucial aspect that affects both the diversity and quality of the generated captions. Various beam sizes were analyzed, such as 1, 3, and 5, with the end determination that a beam size of 3 yielded the most favorable configuration for achieving greater model performance.

The training process utilized A100 and V100 GPUs, providing the computational power required for efficient model training. These GPUs enable accelerated processing, facilitating the optimization of image captioning models.

4.4.6 Training Durations

The duration of the training process varies based on the model architecture and the incorporation of additional linguistic data. For a single LSTM model, the training typically takes approximately 3 days. The 1-1 model, which incorporates a broader dataset encompassing English, Italian, and Spanish, requires around 5 days to complete. Transformer models exhibit a shorter training duration, with a single model completing in approximately 2 days and the 1-1 model in about 3 days. Transformers exhibit superior parameter efficiency in comparison to LSTMs, offering similar or better performance with a reduced number of parameters. This attribute not only accelerates training but also improves computing efficiency.

The adoption of pre-trained models substantially diminishes training time, with a typical

single pre-trained model requiring approximately 1 day, the 1-1 model around 2 days, and the M2 model, incorporating image encoder training, taking about 1 day. In a few-shot setting, completion times are further compressed to just a few hours. The use of pre-trained models, which have learned important patterns from large datasets, speeds up the convergence process, allowing for quick modifications to the specific characteristics of the target data and promoting effective adaptation.

Teacher Forcing during Validation

An important aspect of the training process involves the use of Teacher Forcing during validation. While Teacher Forcing is often used during training to speed up the process, it is important to ensure that validation settings closely resemble real inference conditions. This entails providing ground truth input at every decoding phase, independent of the previously created word. Striking a balance between training efficiency and validation realism is vital for ensuring the model's generalization capabilities.

Tech-stack: The technology stacks used in our research are summarized in Table-4.1, together with their justifications and importance.

Chapter 5

User Study

5.1 Dataset

The development and evaluation of image captioning models hinge on meticulously curated datasets comprising image-caption pairs. These datasets are crucial for enabling the training, validation, and testing stages, serving as a fundamental resource for developing reliable and flexible captioning models. The utilization of benchmark datasets has greatly accelerated research efforts, thereby leading to the progress of state-of-the-art captioning models in the sector. This study evaluates a novel benchmark, the MS-COCO-2014 dataset [55], in three languages: English (the original dataset), Italian, and Spanish (translations of the original dataset).

5.1.1 MS-COCO-2014 Dataset

The MS-COCO dataset [55] is a comprehensive dataset used for tasks such as object detection, segmentation, and captioning. Featuring over 330,000 images, each annotated with 80 object categories and 5 descriptive captions. The MS-COCO-2014 provides a diverse and comprehensive collection of scenes from everyday life. The significance of the MS-COCO dataset lies in its role as a standardized benchmark, fostering advancements in image captioning research by offering a rich, varied, and challenging data for training and evaluating state-of-the-art models. The dataset consists of two primary components: the images and their corresponding annotations. The images are structured in a hierarchical manner, where a main directory encompasses sub-directories for the train, validation, and test sets. And the annotations are represented in JSON format, and includes the following information: train keys: dict_keys(['info', 'images', 'licenses', 'annotations']) [55],

1. **'info':** Information includes essential details like the version number, creation date, and contributor information. Also the url of the official website (e.g., UCI repository page or a distinct domain),


```

info = {
    "year": int,
    "version": str,
    "description": str,
    "contributor": str,
    "url": str,
    "date_created": datetime,
}

```

2. **'licenses'**: The licenses section contains comprehensive information regarding the licenses of the images included in the dataset. This will allow understanding of the precise authorizations given for their utilization. Here is an illustration of licensing information.

```

license = {
    "id": int,
    "name": str,
    "url": str,
}

```

3. **'images'**: This dictionary is considered to be the second most significant one, as it provides metadata about the images.

```

image = {
    "license": int,
    "file_name": str,
    "coco_url": str,
    "height": int,
    "width": int,
    "date_captured": datetime,
    "flickr_url": str
    "id": int,
}

```

where the "license" field represents the ID of the image license, referring to the corresponding entry in the "licenses" section. The "file_name" attribute indicates the name of the file within the images directory. Additionally, "coco_url" and "flickr_url" provide URLs to the online-hosted copies of the image. The "height" and "width" attributes denote the size of the image. Lastly, the "date_captured" field specifies the date when the photograph was taken. The most important field is the "id" field, which is used in "annotations" to identify the image.

4. **'annotations'**: The most important section of the dataset, which contains information vital for each specific tasks like image captioning,

```

annotations= {
    "id": int,
    "image_id": int,
    "caption": str }

```

Here, the "id" corresponds to the unique identifier of the associated image in the dataset. The "image_id" serves as a distinct identifier for the annotation itself, facilitating cross-referencing with 'images' section. Lastly, the "caption" field includes a human-generated description, a vital element for tasks such as image captioning.

The dataset organizes its classes into two main classifications: "things" and "stuff." Within the category of "things," one can discover tangible objects, including animals, automobiles, and household items. Prominent instances of "objects" categories encompass persons, bicycles, cars, and motorcycles. In contrast, the "stuff" category consist of backdrops or environmental elements, including features like sky, water, and highways. The graph in Fig.5.1 below depicts the incorporation of 80 classes in the dataset. From the graph it is crucial to acknowledge that the dataset exhibits class imbalance, where the quantity of samples in one class varies from others. As depicted, the class "person" strongly dominates with 185,316 samples, followed by car and chair, while the class "hair drier" has the fewest samples, totaling only 135. Therefore, the class imbalance might lead to a potential bias during training and evaluation phase of the model. This bias, in turn, may result in over-fitting to the majority class, leading to excellent performance within that class but subpar performance in other classes

In addition to class imbalance, the COCO dataset exhibits variability in image dimensions, encompassing a total of 2,519 distinct dimensions. Among these, the most frequent dimension is (640, 480), observed in 26,464 images. The largest image in the dataset possesses dimension of (640, 640), whereas the smallest image is considerably more compact with dimensions of (59, 72). Understanding this diversity is crucial for image captioning models that necessitate consistent dimensions, making it pivotal in deciding on a standardized input size for pre-processing.

5.1.2 MS-COCO-it Dataset

The MS-COCO-it [67] dataset originates from the study titled "Large scale datasets for Image and Video Captioning in Italian," accessible at [it_dataset](#). Comprising over 600,000 image-caption pairs, it is an Italian-translated version of the original English MS-COCO dataset. Following the methodology outlined by Vinyals et.al. [78], the image-caption pairs were utilized for training, excluding a development set of approximately 2000 images and a test set of around 4000 images. These sets are termed the MS-COCO2K development set and the MS-COCO4K test set. Each image is accompanied by five captions, which have been automatically translated from English to Italian and subsequently manually validated. The MS-COCO-it dataset mirrors the format and images of the original MS-COCO dataset, comprising both unvalidated (u.) and validated (v.) elements. Table-5.1 encompasses the size of training, validation and test data.

	Image	Captions	Words
training u	116,195	581,286	≈ 6,900,000
development v.	308	1,541	17,913
development u.	1,696	8,486	≈ 102,000
test v.	596	2,982	34,657
test u.	3,422	17,120	≈ 202,000

Table 5.1: Data Overview: MSCOCO-it Dataset

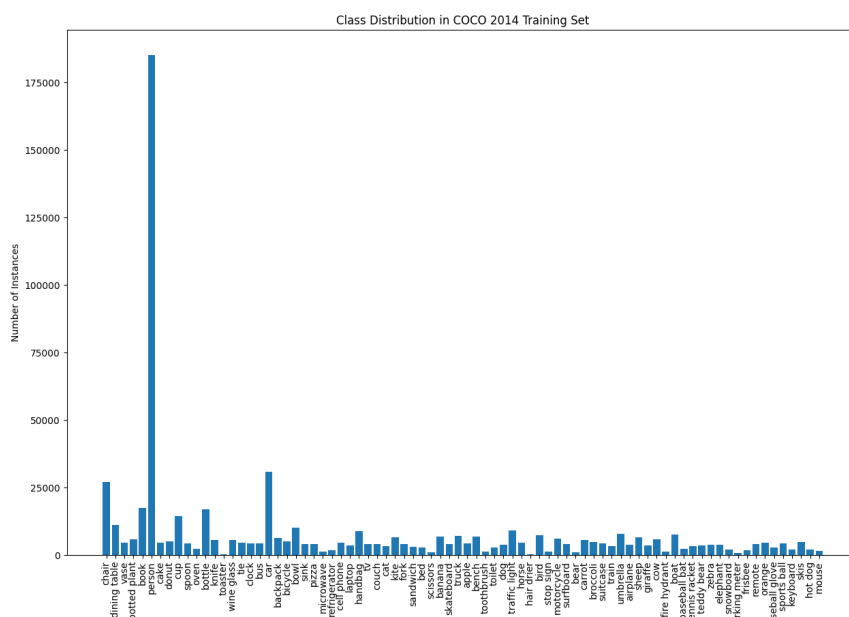


Figure 5.1: Distribution of Data Classes for the MS-COCO dataset: Illustrating the relative frequencies of different classes within the dataset

	Image	Captions
train_human_spanish	2,000	10,000
train_machine_spanish.	20,000	100,000
validation	1,000	5,000
test	1,000	5,000

Table 5.2: Data Overview: MSCOCO-es Dataset

5.1.3 MS-COCO-ES Dataset

The MS-COCO-ES dataset, introduced in [5], is a derivative of the original MS-COCO dataset, crafted through the process of translation. The primary objective of the project was to present a streamlined subset of the initial image captions, skillfully translated into Spanish by human annotators. This curated subset comprises 20,000 captions corresponding to 4,000 images, maintaining an average of 5 captions per image. Table-5.2 sums up the size of the Spanish dataset.

5.1.4 Comparison between English, Italian and Spanish dataset

Notably, each image is accompanied by five English captions, each offering nuanced insights. Fig.5.2, Fig.5.3 and Fig.5.4 provides an example for English, Italian and Spanish dataset, respectively. The annotations aim to capture the richness and diversity of the dataset. The data sizes for all three languages are compared in Table-5.3.

The English captions (see Fig.5.5) can range from a minimum of 5 words to a maximum

Table 5.3: Summary of Images and Annotations Statistics: displaying the number of images and corresponding captions for various languages (English, Italian and Spanish)

		English	Italian	Spanish
Images	Training	82783	82080	22000
	Validation	40503	34115	5000
Annotations	Training	414113	410596	110000
	Validation	202654	170690	1000



1. A large bear laying on the side of a rocky
2. A black bear sitting on rock with mossy patches.
3. A brown bear shows his claws while lazing at the zoo
4. A close up of a bear laying on a large rock
5. A large bear that is laying down on a rock.

Figure 5.2: An example image from English COCO dataset

of 49 words. Italian captions (see Fig.5.6), on the other hand, might span from 6 to 55 words, while Spanish captions (see Fig.5.7) can have a range of 2 to 59 words. The English dataset shows that captions with 10 words have the highest frequency, suggesting a common preference among annotators. Conversely, captions consisting of 5 words have the lowest frequency of occurrence. In contrast, the Italian dataset shows a significant rise in the frequency of captions with 11 words, while those with 6 words are the least common. Similarly, the Spanish dataset reveals that captions consisting of 10 words are the most prevalent, whilst captions with only 2 words have the lowest occurrence. This dissertation contributes to the development of language models and improves the efficiency of the algorithms, resulting in enhanced image captioning tasks.

5.2 Evaluation Metrics

Evaluating the efficacy of the implemented models demands an evaluation of the output caption's quality. In the study we employ BLUE (Bilingual Evaluation Understudy)



1. La carrozza trainata da cavalli si muove lungo la strada trasportando due passeggeri
2. Una coppia seduta sul retro di una carrozza trainata da cavalli.
3. Un giro in carrozza e cavalli in una città vecchia
4. La gente cavalca su una carrozza trainata da cavalli con ruote gialle.
5. Un uomo sta dando gite in carrozza a una coppia

Figure 5.3: An example image from Italian COCO dataset



1. Un autobús rojo de dos pisos con palmeras al fondo.
2. Un autobús de dos pisos descolorido y rojo circulando entre las palmeras.
3. Un autobús de dos pisos por una carretera con palmeras.
4. Un viejo autobús de dos pisos en la calle.
5. Un gran autobús rojo en una calle pavimentada.

Figure 5.4: An example image from Spanish COCO dataset

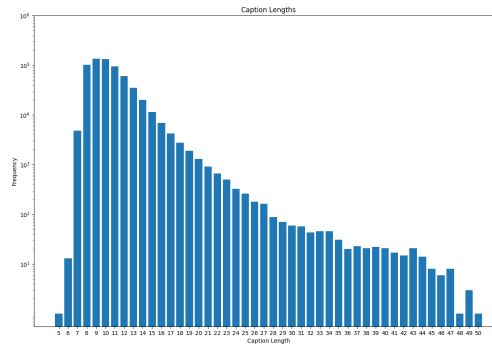


Figure 5.5: Distribution of Caption Length in English MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 5 words to a maximum of 49 words

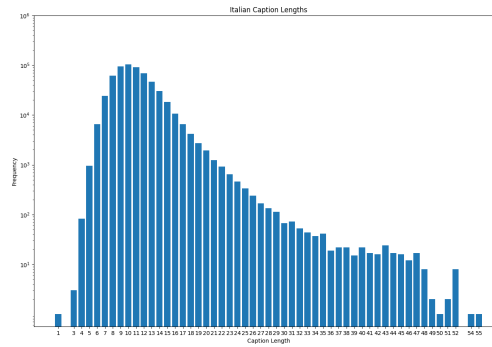


Figure 5.6: Distribution of Caption Length in Italian MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 6 words to a maximum of 55 words

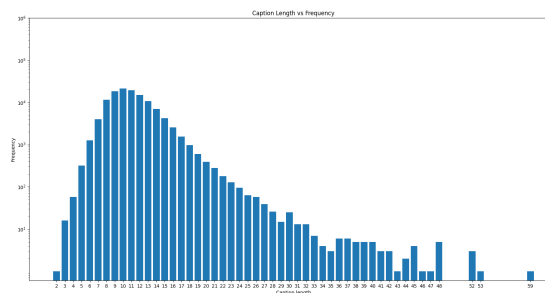


Figure 5.7: Distribution of Caption Length in Italian MS COCO Dataset: This bar graph visualizes the varied lengths of captions ranging from a minimum of 2 words to a maximum of 59 words

[62], ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [54], and CIDEr (Consensus-based Image Description Evaluation) [77] to provide an in-depth understanding of the models' performance. By employing these metrics, the aim is to assess the quality of the captions by considering variables such as accuracy, completeness, and linguistic similarity.

5.2.1 n-grams

To begin, let's establish the concept of "n-grams," which refers to consecutive sequences of 'n' elements, typically words or characters, extracted from a corpus of texts. N-grams entail predicting the likelihood of a word occurring, considering the context provided by the preceding n-1 words. The choice of "n" determines the scope of the context taken into account by the model. For instance, n=1 corresponds to unigrams (individual words), N=2 is bigrams, and N=3 corresponds to trigrams and so on. Given a sentence "The cat is sitting on a mat", the n-grams are as follows,

1. 1-gram (unigram): "The", "cat", "is", "on", "a", "mat"
2. 2-gram (bigram): "The cat", "cat is", "is on", "on a", "a mat"
3. 3-gram (trigram): "The cat is", "cat is on", "is on a", "on a mat"
4. 4-gram: "The cat is on", "cat is on a", "is on a mat" and so on.

N-grams offer a direct and effective approach to capturing contextual information and connections between words in a sequence. However, they encounter limitations such as the issue of sparsity, which hinders their ability to make generalizations because of rare or unseen sequences. The limited context window restricts the ability to capture distant relationships, and resolving ambiguity continues to be difficult due to the lack of complex semantic comprehension in N-grams. Subword-level N-grams help to reduce the out-of-vocabulary problem.

5.2.2 BLEU (Bilingual Evaluation Understudy)

BLEU score [62] is a quantitative metric employed for comparing a predicted caption with one or many reference captions. The BLEU score is a precision-oriented metric that ranges from 0 to 1. A score of 0 signifies that the created output lacks any resemblance to the references, suggesting subpar caption generation. On the other hand, a score of 1 indicates that the created output matches the references perfectly, showing a high level of caption quality. To compute the BLEU score, we first determine the modified precision and brevity penalty (BP).

Modified Precision

Precision is defined as the proportion of true positive predictions among all positive predictions. In brief, it quantifies the accuracy of the model's positive predictions. Within the framework of n-grams, true positives refer to the set of n-grams that match exactly between the predicted and reference sentences. Additionally, false positives relate to the n-grams that are present in the predicted sentences but not in the reference sentence. Hence, precision can be determined by dividing the number of matched n-grams by the total number of n-grams in the predicted sentences. Let's consider the following example:

1. Reference: "The cat is sitting on the mat"
2. Prediction: "The cat cat cat mat"

What is the degree of accuracy in this scenario when examining individual words (unigrams)? Nevertheless, it is clear that this translation is insufficient. To address this issue, the computation of BLUE involves the application of "Modified Precision."

To calculate modified precision, the frequency of an n-gram is clipped to the highest number of times it occurs in the reference texts. The modified n-gram precision can be precisely defined as follows:

$$P = \exp\left(\sum_{n=1}^N w_n \log(p_n)\right) \quad (5.1)$$

where $w_n = 1/n$.

Brevity Penalty(BP)

In instances where longer sentences are involved, it's possible that the generated candidates may be notably concise and may lack essential details when compared to the reference. In order to tackle this issue, the notion of the Brevity Penalty (BP) is introduced. The Brevity Penalty is utilized to penalize predictions that are excessively brief in comparison to their corresponding references. This approach guarantees that the generated content will possess a similar level of completeness and appropriateness as the specified references.

$$BP = \begin{cases} 1, & \text{if } c > r, \\ \exp(1 - r/c), & \text{otherwise,} \end{cases} \quad (5.2)$$

where c is the predicted caption length and r is the average length of the ground-truth caption.

BLEU Score

The geometric mean of the modified precision up to N is multiplied by the BP value to obtain the final BLEU score.

$$BLEU = BP * P \quad (5.3)$$

The BLEU score has multiple advantages, such as its straightforward calculation and extensive acceptance as an evaluation metric. It has demonstrated a strong correlation with human judgment. However, it is imperative to consider notable disadvantages. The score does not take into account semantic meaning when evaluating synonyms of n-grams unless they exactly match the references. For instance, multiple expressions with similar meanings in a particular text might lead to a poor blue score. In addition, higher-order n-grams face difficulties in addressing word order problems, which might result in misleadingly high ratings for translations with bad word order. Cross-dataset comparisons are hard to do when using different methods because things like the number of references and the normalization and tokenization techniques employed have a big effect on BLEU scores.

5.2.3 ROUGE (Recall Oriented Understudy for Gisting Evaluation)

ROUGE [54] measures the extent of overlap and similarity between the predicted captions produced by the model and the reference captions. The evaluation considers various aspects like the overlap of n-grams, the overlap of words, and the capacity to recall important phrases. ROUGE scores are calculated using precision, recall, and F1-score metrics, and they assist in evaluating the quality of the predicted caption by comparing it to the reference phrase. ROUGE can be categorized into different types, such as ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S, based on the particular feature used for calculation.

1. ROUGE-N: The ROUGE-N metric quantifies the degree of similarity between the predicted caption and the ground truth caption by evaluating the overlap of n-grams [53]. For example, ROUGE-1 (unigram) quantifies the degree of similarity between individual words, ROUGE-2 (bigram) quantifies the degree of similarity between two-word sequences, and so forth. The ROUGE-N metric is employed to assess the fluency and grammatical accuracy of the generated captions. The precision, recall, and F1-score are computed by taking into account the overlap of n-grams.

$$Recall = \frac{Overlapping\ number\ of\ n - grams}{Number\ of\ n - grams\ in\ the\ reference\ sentence} \quad (5.4)$$

$$Precision = \frac{Overlapping\ number\ of\ n - grams}{Number\ of\ n - grams\ in\ predicted\ sentence} \quad (5.5)$$

The F-1 score is computed as (Harmonic mean),

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)} \quad (5.6)$$

2. ROUGE-L: The metrics ROUGE-L are calculated based on the ideas of Longest Common Subsequence (LCS) [54]. ROUGE-L is specifically designed for the purpose of evaluating the Longest Common Subsequence (LCS). The Longest Common Subsequence (LCS) is the sequence of words that appears in both the candidates and reference summaries, with the requirement that the words must maintain their original order. It automatically includes the longest uninterrupted sequence of words. It is important to recognize that LCSes may not be consecutive, yet they remain in same order. For example:

(a) Predicted sentence: "The cat is sleeping on the red mat"

(b) Reference sentence: "The cat is lying down on a clean red mat"

To calculate ROUGE-L, the first step is to identify the longest common subsequence that is shared by the two phrases. This example illustrates the phrase "The cat is on the red mat". Subsequently, calculate the precision, recall, and F1-score.

$$P = \frac{LCS(A, B)}{m} \quad (5.7)$$

$$R = \frac{LCS(A, B)}{n} \quad (5.8)$$

where, A and B represent predicted and reference captions, which have respective lengths of m and n. Then F-score is calculated as the weighted Harmonic mean as,

$$F = \frac{(1 + b^2)RP}{R + b^2P} \quad (5.9)$$

ROUGE stands as a robust metric with demonstrated correlation with human evaluation, providing a reliable means of assessing the quality of generated summaries. The simplicity of its computation and understanding contributes to its extensive acceptance, and its ability to be used with other languages makes it suitable for evaluating summaries in multiple languages. Nevertheless, ROUGE's main emphasis on n-gram overlap leads to limits by disregarding the semantic intricacies inherent in the meaning of the summary. Moreover, the metric's sensitivity to the selection of reference summaries creates heterogeneity in the evaluation process. Moreover, ROUGE may display partiality for summaries that have a different length compared to the reference summaries.

5.2.4 CIDEr (Consensus-based Image Description Evaluation)

The evaluation of the textual descriptions generated for images can be conducted using the CIDEr metric [77]. The CIDEr measure evaluates the similarity between a generated caption and the reference captions by considering not just word choice and grammar but also meaning and content. The CIDEr measure calculates how similar a generated caption is to the reference captions.

CIDEr computes similarity by considering the common n-grams (phrases of different lengths) between the generated caption and the reference captions. It focuses on collecting many methods of presenting the same fundamental ideas, which is very valuable for evaluating the depth and fluency of created captions. The calculation of the CIDEr metric comprises multiple stages:

1. Begin by providing a set of reference captions for each image, establishing these captions as the ground truth for the evaluation process.
2. Compare the generated caption to each reference caption using the BLEU (Bilingual Evaluation Understudy) score. This metric evaluates the overlap of n-grams between the generated caption and the reference captions.
3. Modify the BLEU scores through IDF (Inverse Document Frequency) weighting. This adjustment assigns more significance to words that are infrequent in the reference captions but appear in the generated caption.
4. Conclude the process by averaging the weighted BLEU scores across all reference captions, yielding the ultimate CIDEr score. This comprehensive approach considers both n-gram overlap and word rarity, providing a nuanced evaluation of the quality of generated captions

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(S_{ij})}{||g^n(c_i)|| ||g^n(S_{ij})||} \quad (5.10)$$

$$CIDEr(c_i, S_i) = \sum_{n=1}^N w_n CIDEr_n(c_i, S_i) \quad (5.11)$$

Models	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
CNN-LSTM	74.00	57.23	43.52	33.11	51.96	52.61	1.033
CNN-Transformer	74.92	58.59	44.14	34.81	53.11	53.69	1.058
CNN-Marian NMT Decoder	75.81	64.73	56.72	47.85	61.27	55.00	1.353
CNN-Modified Marian NMT Decoder	76.08	65.73	56.04	48.17	61.50	56.08	1.408

Table 5.4: Performance Metrics English caption Generation by a single model: A Comparative Analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various models

The CIDEr metric has gained widespread adoption in the domain of image captioning and has been employed in numerous benchmark datasets and competitions. The evaluation metric is extensively utilized due to its ability to give a comprehensive assessment of the quality of generated captions, considering both the linguistic and content aspects.

5.3 Results

Here, we present the comprehensive results of our approach for generating descriptive captions for an image. The review covers both quantitative and qualitative factors, providing insights into the performance of three approaches: the Single model, the 1-1 model, and the M2 model. The following analysis explores the complexities of different architectures, as explained in Section.4.1. Following that, a thorough analysis of various elements within our model is conducted, offering an informative and comprehensive assessment. Our research not only sheds light on the effectiveness of our image captioning models but also provides a detailed understanding of the specific contributions of different architectural components. The examination of results will be directed by the research question stated in Section.1.3. As we examine the results, our analysis will focus on answering the particular hypotheses that were raised at the beginning of our research. This systematic strategy guarantees a concentrated and logical examination of results in accordance with the overarching objectives specified in our research question.

5.3.1 Research Question: 01

How does performance differ across three different decoder architectures: LSTM, Transformer, and Pre-trained NMT Transformer?

Our research involves examining different configurations of the encoder-decoder framework, each providing varied combinations of components. These configurations include CNN-LSTM, CNN-Transformer, CNN-pretrained Transformer, and CNN-modified pre-trained Transformer. Section.4.1 offers a comprehensive explanation of these models, clarifying their architectural complexities. We will mostly concentrate on comparing the results acquired from these different models in order to identify the differences in performance and the specific strengths. This comparative evaluation sets the foundation for a nuanced understanding of the impact of different encoder-decoder combinations on the image captioning task.

Quantitative Analysis:

Models	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
CNN-LSTM	70.92	56.62	43.13	33.37	51.01	50	0.944
CNN-Transformer	71.13	56.03	44.50	34.36	51.50	51.29	0.96
CNN-Marian NMT Decoder	71.44	62.86	54.71	47.79	59.20	53.62	1.079
CNN-Modified Marian NMT Decoder	73.94	63.16	55.98	48.04	60.27	54.62	1.191

Table 5.5: Performance Metrics for Italian caption Generation by a single model: A Comparative Analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various models

Table 5.4 and 5.5 provide a detailed breakdown of performance metrics, including BLEU-1, BLEU-2, BLEU-3, BLEU-4, Avg-BLEU, ROUGE-L, and CIDEr scores, for both English and Italian datasets. Noteworthy observations emerge from the analysis of Table.5.4, focusing on the English dataset. The modified pre-trained transformer model demonstrates superior performance compared to other models, particularly evident in the BLEU-4 score, which attains 48. This represents a notable relative improvement of +37% over the LSTM model and +34% over the non pre-trained transformer model. This significant enhancement is attributed to harnessing the language understanding capabilities of the pre-trained model, resulting in a +2% relative improvement over the Marian NMT decoder. The efficacy of the modified pre-trained transformer model is further emphasized by its performance across various evaluation metrics, with trends similar to those observed for BLEU-4. The benefits of this model are accentuated by the intrinsic characteristics of the 1x1 convolution, especially pronounced when processing image inputs. Moving beyond BLEU-4, consistent trends across other evaluation metrics underscore the robustness and efficacy of the proposed model.

In Table.5.5 mirrors the trends observed in Table.5.4, albeit for the Italian dataset. The modified pretrained transformer model consistently outperforms the non pre-trained transformer model by +34%, exhibiting a +2% relative gain over the Marian NMT model. These consistent patterns underscore the robustness and generalizability of the pretrained transformer approach across different linguistic datasets. This collective evidence substantiates the effectiveness and versatility of the proposed model, further reinforcing its potential applicability in diverse language-processing scenarios.

The experimental research revealed that the LSTM model requires around three days to complete the training process. It achieves convergence at 26th epochs before showing signs of overfitting. On the other hand, the Transformer model demonstrates accelerated convergence, completing training in just two days and achieving up to 19th epochs before showing signs of overfitting. The effectiveness of the Transformer architecture can be attributed to its increased number of inherent parameters. The use of a pretrained model, notably the Marian Neural Machine Translation (NMT) model, significantly reduces the time required for training, completing in just one day and showing evidence of overfitting as early as the 15th epoch. Likewise, a modified pre-trained model achieves convergence in just 14th epochs, highlighting the significant benefits of utilizing existing information to improve the efficiency of the model training process. It is important to mention that, in all of our experiments, we consistently use a batch size of 128.

Models	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
CNN-LSTM	43.61	34.69	28.01	23.66	32.49	34.63	0.69
CNN-Transformer	44.46	36.39	30.35	25.49	34.17	36.17	0.7551
CNN-Marian NMT Decoder	48.25	42.37	32.84	28.46	37.98	38.36	0.7932

Table 5.6: Performance Metrics for English caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models

Models	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
CNN-LSTM	42.19	34.81	28.72	23.68	32.35	33.62	0.65
CNN-Transformer	43.36	36.58	30.25	25.18	33.84	35.60	0.7534
CNN-Marian NMT Decoder	48.02	41.58	31.76	27.11	37.11	38.70	0.7765

Table 5.7: Performance Metrics for Italian caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models

5.3.2 Research Question: 02

How does the 1-1 model performance differ across three different decoder architectures: LSTM, Transformer, and Pre-trained NMT Transformer?

After examining the data in Table.5.6, it is clear that the pre-trained Neural Machine Translation (NMT) model performs significantly better than other models in the 1-1 model approach, specifically in relation to the English language. This superiority is demonstrated by an increase of 18% compared to the LSTM model and 11% compared to the non pre-trained Transformer model. Table 5.7 demonstrates similar findings for the Italian language. The pre-trained model surpasses other models by achieving a relative improvement of +13% compared to the LSTM model and +7% compared to the Transformer model.

Regarding the Spanish language, as explained in Table 5.8, the consistent pattern continues, highlighting the superior performance of the pre-trained model with a relative improvement of 36% compared to the LSTM model and 32% compared to the Transformer model. The consistency of these results across different languages emphasizes the effectiveness of the pre-trained NMT model in the 1-1 strategy, thus demonstrating its significant performance benefits compared to other models.

After comparing the datasets in the three languages, it is evident that English and Italian produce similar outcomes, with BLEU-4 ratings of 28.46 and 27.11, respectively. On the other hand, the Spanish dataset shows a relatively lower level of effectiveness, with a BLEU-4 score of 8.84. The discrepancy might be ascribed to the utilization of shared model parameters across the languages. In addition, the datasets for English and Italian are larger than the Spanish dataset, which leads to its reduction. The disparities in dataset sizes and parameter distribution jointly contribute to the reduced performance of the model on the Spanish dataset compared to the English and Italian datasets.

We employed the Lang_detect library [8] within the Python programming language to validate that the generated caption is same as that of the target languages. The results of this validation are presented in Table 5.9, illustrating the comparative analysis be-

Models	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
CNN-LSTM	19.27	11.75	8.16	6.10	11.32	16.32	0.2014
CNN-Transformer	19.06	12.79	8.36	6.36	11.64	16.42	0.2079
CNN-Marian NMT Decoder	24.26	17.69	10.25	8.84	15.26	18.55	0.2212

Table 5.8: Performance Metrics for Spanish caption Generation by a 1-1 model: A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr scores across various models

	Actual Language				Total
		English	Italian	Spanish	
Predicted Language	English	24,724	276	0	25,000
	Italian	102	24,587	311	25,000
	Spanish	9	126	4865	5000

Table 5.9: The above table illustrates the number of times the model correctly/incorrectly generated the caption when it was explicitly instructed about the target language

tween the total number of inference samples and the corresponding accurately captioned instances in the target languages. The assessment of the 1-1 model is undertaken independently for each language. For the English dataset, which encompasses a total of 25,000 captions, our model accurately captioned 24,724 instances in English. However, 276 captions were erroneously classified as Italian. In the case of the Italian dataset, comprising 25,000 examples, our model accurately captioned 24,587 instances as Italian. Nonetheless, 102 captions were inaccurately labeled as English, and 311 as Spanish. Finally, for the Spanish dataset consisting of 5,000 examples, 4,865 captions were correctly labeled as Spanish. Nevertheless, 9 captions were misclassified as English, and 126 as Italian.

5.3.3 Research Question: 03

Compare the performance of pre-trained NMT model with the modified pre-trained NMT model

Quantitative analysis

Figure 5.8 and Fig.5.9 offers valuable insights into the performance difference between a model using a 1x1 convolution layer and a model using a fully connected layer. The pre-trained Marian Neural Machine Translation (NMT) decoder has been utilized in this experimental work. The empirical results clearly demonstrate that the model with a 1x1 convolution layer achieves a much faster convergence rate compared to the model using a linear layer. More precisely, the evaluation loss reaches its lowest value of 1.1 after around 7,000 training steps for the model with 1x1 convolutional layer. On the other hand, the fully connected layer model achieves a loss of 1.4 but takes approximately 10,000 steps to converge. The noticeable discrepancy in the speed at which the model converges suggests that including a 1x1 convolution layer improves the efficiency of the training process.

The integration of the 1x1 convolutional layer in our model is grounded in its capacity to offer substantial advantages. The 1x1 convolutional layer primarily utilizes fewer



Figure 5.8: Graph depicting evaluation loss for pre-trained MarianNMT model with 1x1 convolution on the English MS-COCO dataset



Figure 5.9: Graph depicting evaluation loss for pre-trained MarianNMT model on the English MS-COCO dataset

parameters compared to the fully connected layer. It operates on local receptive fields rather than forming connections between every input and every neuron. The decrease in the number of parameters allows for a more efficient learning process, resulting in the model converging quickly during training. The convolutional layer is able to effectively utilize correlations within the input by capturing spatial hierarchies and local patterns, surpassing the capabilities of a fully connected layer. The subsequent decrease in loss implies that the model with the 1x1 convolutional layer performs better on new data, suggesting a higher capacity to extract and understand significant features from the input. The inherent advantages of the 1x1 convolutional layer, such as its efficient use of parameters and improved extraction of features, result in more rapid convergence during training and higher performance in reducing evaluation loss.

Qualitative Analysis

In the realm of qualitative analysis, a closer examination of Fig.5.10 unveils the model's adeptness in directing its attention to the pertinent regions of the image during caption generation. Observing the process, it becomes evident that the model strategically focuses on the appropriate elements within the image for accurate captioning. Noteworthy instances include the generation of the word "couple," where the model directs its attention to both buses, demonstrating a nuanced understanding of the scene. Similarly, when conjuring the word "down," the model zeroes in on the lower section of the bus, aligning seamlessly with the intended focus. In essence, the model's capability to maintain context-aware attention, ensuring that the generated captions accurately encapsulate the salient features of the depicted scenes. The incorporation of 1x1 convolutions, fostering cross attention, emerges as a pivotal factor in enhancing the model's focus and overall performance during the caption generation process.

Examining Fig.5.11, a discerning observation reveals a noteworthy aspect of the model's caption generation process—it appears to lack precision in directing attention to the appropriate elements within the image. The generated captions seem to scatter focus indiscriminately, lacking a cohesive connection to the visual content. This raises a compelling inference: the model, as depicted in the figure, may not be effectively leveraging information from the image encoder during the captioning process. Instead, it seems to heavily rely solely on the language decoder, potentially resorting to a form of rote memorization. The scattered focus observed in the generated captions suggests that connection between the visual and linguistic components is lacking. Consequently, the model's reliance on the language decoder without effectively incorporating information from the image encoder raises concerns about its depth of understanding about the image features.

After a careful analysis of the generated captions, a clear distinction becomes apparent between Fig.5.10 and Fig.5.11. Significantly, the captions derived from Fig.5.10 demonstrate a high level of grammatical fluency in comparison to those from Fig.5.11. The difference in linguistic coherence suggests a strong conclusion: the modified pre-trained model, depicted in Fig.5.10, seems to gain substantial advantages from the inclusion of 1x1 convolution layers. The impact of these layers is clearly obvious in the improved linguistic quality of the generated captions, demonstrating that the use of convolutional methods has a favorable effect on the model's ability to generate language. Essentially, this comparison highlights the crucial function of 1x1 convolution layers in enhancing the language creation skills of the pre-trained model. The noticeable enhancement in grammatical proficiency is evidence of the effectiveness of these structural changes in maximizing the model's performance for generating captions.

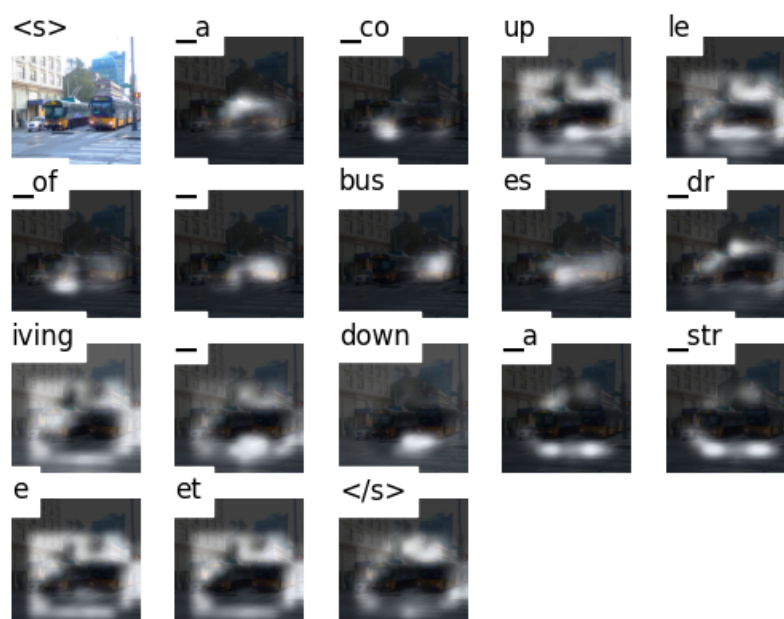


Figure 5.10: Analyzing CNN encoder-modified pretrained MarianNMT transformer: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

5.3.4 Research Question: 04

Compare the performance between single model, 1-1 model and M2 model?

Quantitative analysis

In this comparative analysis, the efficacy of three distinct methodologies is assessed: the Single model, 1-1 model, and M2 model. The outcomes, as delineated in Table 5.10, underscore notable differentials in performance metrics. The evaluation is conducted on the Spanish dataset utilizing the pre-trained Neural Machine Translation (NMT) decoder. Notably, the M2 model emerges as a preeminent performer, exhibiting a commendable +11% relative enhancement compared to the Single model. The respective BLEU-1 scores for the M2 model and Single model are 74.50 and 66.68. Upon closer scrutiny, the M2 model manifests an exceptional +25% relative improvement over the Single model and a substantial advancement over the 1-1 model, particularly discernible in the BLEU-4 score. This discerning analysis sheds light on the superior performance of the M2 model, emphasizing its efficacy in comparison to alternative approaches.

The suboptimal performance of the 1-1 model can be attributed to its training on a composite dataset encompassing English, Italian, and Spanish concurrently. Although this approach is ambitious, it leads to a capacity bottleneck and diminished maintainability. The inherent constraint in the parameter size of the pre-trained model impedes its ability to effectively learn all three languages simultaneously. This limitation prompts



Figure 5.11: Analyzing CNN encoder-pretrained MarianNMT transformer: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

consideration of larger models, such as Lamma [74], which boasts a substantial 7 billion parameters and holds the potential to yield superior results. Adding to the challenges, the 1-1 model faces inherent difficulty due to the lower volume of the Spanish dataset compared to English and Italian, further contributing to its suboptimal performance. On the other hand, the Single model stands out by focusing exclusively on learning one language, even though it has limited parameters. In addition, the Single model utilizes transfer learning by incorporating knowledge from the Neural Machine Translation (NMT) decoder, hence improving its overall efficacy. The M2 model introduces an innovative approach with cross-lingual transfer learning, where both the encoder and decoder undergo pre-training. This unique design allows the image encoder to seamlessly adapt to the language decoder, facilitating the sharing of language-specific decoders. The success of the M2 model is attributed to its effective utilization of pre-trained decoders, highlighting the essentiality of leveraging pre-trained components to get exceptional performance in multilingual tasks.

Ultimately, the M2 model surpasses both the 1-1 model and the Single model when it comes to cross-lingual image captioning. Nevertheless, it is important to remark that the 1-1 model possesses the advantage of concurrently training on three languages and demonstrates favorable performance for all three languages in term of fluency and accuracy of the captions, as stated in Research Question.5.3.2. While the M2 model has shown its ability to adapt, making changes to the 1-1 model, such adding a new language, poses significant difficulties. Incorporating another language requires a thorough retraining

	BL1	BL2	BL3	BL4	AVG-BL	R-L	CDr
Single Model	66.68	55.97	47.55	40.28	52.62	51.41	1.0183
1-1 Model	24.26	17.69	10.25	8.84	15.26	18.55	0.2212
M2 Model	74.50	66.11	58.76	52.07	62.85	56.44	1.2043

Table 5.10: Performance Metrics comparing the performance between Single model, 1-1 model and M2 model: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.

of the entire model as a cohesive unit, which requires a significant commitment of both time and effort. Experiments conducted with the Transformer and LSTM frameworks confirm this claim. When trained on a combined dataset of English and Italian, the LSTM model took around 3 days to reach convergence, whereas the Transformer model reached convergence in just 2 days. Yet, the complexity increases when dealing with a dataset that includes three languages: English, Italian, and Spanish. Within this particular framework, the LSTM model had a convergence period of approximately 5 days, whereas the Transformer model required nearly 4.5 days to achieve convergence. This significant increase in the period of instruction highlights the lack of feasibility in solely enhancing languages to the 1-1 approach. The significant amount of time and resources required for retraining emphasizes the need for more effective and adaptable approaches in the development of multilingual models.

Qualitative Analysis

The visual analysis of Fig.5.12, Fig.5.13, and Fig.5.14 provides valuable insights into the language generation capabilities of different approaches. Evidently, the single model, depicted in Fig.5.12, stands out by producing more fluent sentences compared to its counterparts. This proficiency can be attributed to its training approach, focusing solely on one language, namely Spanish. The absence of parameter sharing with other languages allows this model to tailor its linguistic nuances precisely, resulting in enhanced sentence fluency. In stark contrast, the 1-1 model, as illustrated in Fig.5.13, exhibits a suboptimal performance. The shared parameters across three distinct languages compromise its ability to generate fluent sentences. Additionally, the model's relatively smaller size for a 1-1 configuration may contribute to its diminished capability, highlighting the significance of appropriate model size in linguistic tasks. Turning attention to Fig.5.14, the M2 model displays commendable caption generation in a few-shot setting, having been trained on a mere 2000 Spanish samples. This success can be attributed to its utilization of cross-lingual transfer learning, allowing the model to leverage knowledge from multiple languages efficiently.

In summary, the visual evidence underscores the impact of training strategies and model architectures on language generation. The single model excels due to its language-specific training, while the 1-1 model faces challenges with shared parameters and a relatively smaller size. The M2 model's success in a few-shot setting attests to the efficacy of cross-lingual transfer learning in harnessing linguistic proficiency from limited training samples. Further illustrations showcasing the modified NMT transformer are provided in the Appendix section. The redesigned NMT transformer generates captions that are not only more precise but also demonstrate enhanced fluency.

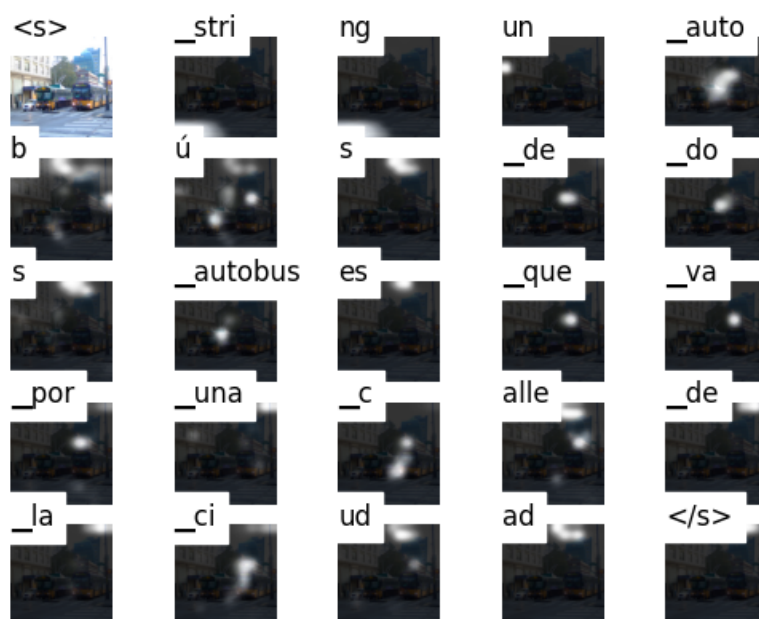


Figure 5.12: Analyzing CNN encoder-pretrained MarianNMT Transformer for Single model approach: Saliency map depicting the model’s inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

5.3.5 Research Question: 05

Can adapting the M2 model to the image captioning task enable few-shot image captioning?” (with Marian NMT Decoder)

Quantitative analysis

The empirical results obtained from studies with various sample sizes reveal a noticeable trend in the model’s performance. There is a noticeable decrease in the overall model performance as the amount of training data decreases. Table.5.11 demonstrates the observed pattern, where utilizing 22,000 samples for few-shot learning results in a BLEU-4 score of 52. It is important to note that, even when the sample size is reduced to 2,000, the BLEU-4 score remains constant at 52. Nevertheless, when the sample size is reduced to 1,000, the BLEU-4 score decreases to 48. Further reduction to 500 samples yields a score of 42. A further experiment employing zero-shot learning, without any training data, produces a score of 5. Due to the structural similarities between Italian and Spanish languages, the model is able to accurately predict some common tokens in zero-shot learning, resulting in a BLEU-1 score of 34. These data emphasize how the model’s performance is affected by different sample sizes and demonstrate the significant effect on its ability to make accurate predictions, especially in situations involving languages with limited resources. In the domain of few-shot learning, a similar trend may be observed with the modified pre-trained Neural Machine Translation (NMT) model. Table.5.12 shows that the BLEU score declines continuously as the number of samples

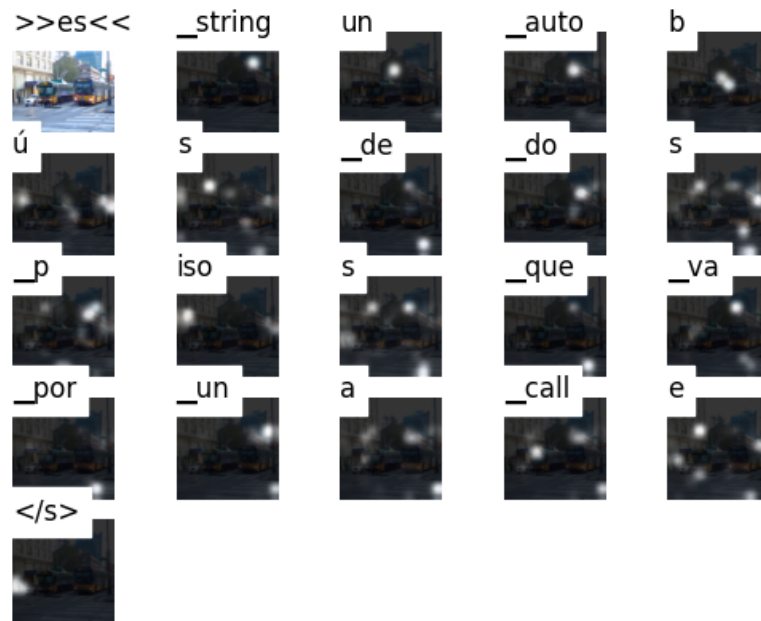


Figure 5.13: Analyzing CNN encoder-petrained MarianNMT Transformer for 1-1 model approach: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

decreases, suggesting a clear relationship between the size of the training data and the model's performance.

After comparing Table.5.11 and Table.5.12, it is evident that the pre-trained NMT model performs better than the modified pre-trained NMT model. The pre-trained NMT model shows a relative performance decline of -12% compared to the modified pre-trained NMT model, using a complete dataset of 22,000 samples. Nevertheless, when the sample size is decreased to 500, the pre-trained NMT model demonstrates a relative performance improvement of +8%.

The limitations of the 1-1 model are apparent in its inability to execute few-shot learning tasks, primarily attributable to the model's constrained size and limited capacity. The selected model lacks the necessary parameters to adeptly adapt to the demands of few-shot learning. A potential avenue for improvement arises by opting for a larger model equipped with billions of parameters. The M2 model is characterized by its parameter-sharing method between different modules, which facilitates the flow of information in an inter-lingua space. This novel technique enhances the model's flexibility and overall effectiveness. In addition, the M2 model exhibits a significant improvement in performance when applied to low-resource language pairs. The cross-linguistic impact becomes apparent when multiple languages are incorporated into a unified module. Significantly, low-resource languages demonstrate improved performance when trained together with high-resource language pairings inside the same module. The utilization

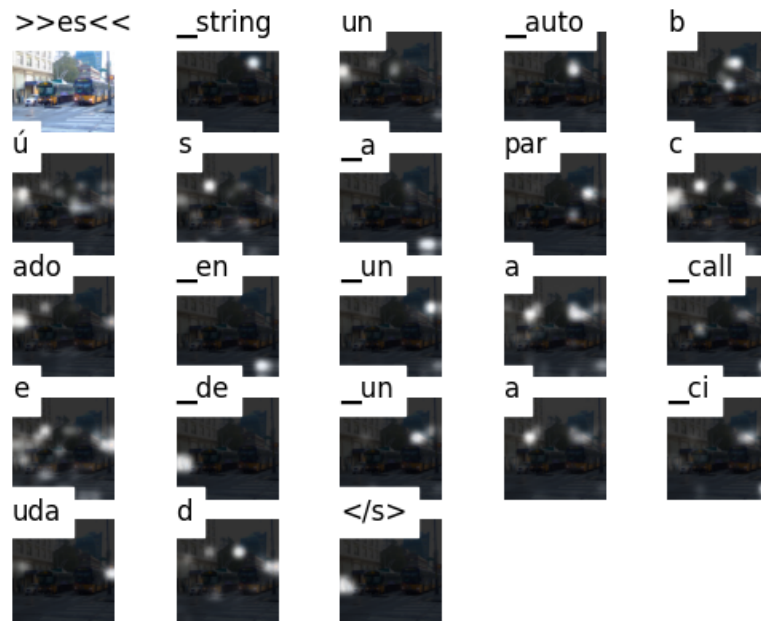


Figure 5.14: Analyzing CNN encoder-pretrained MarianNMT Transformer for M2 architecture: Saliency map depicting the model's inference on 2000 samples Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

of this multi-way training technique offers significant benefits, especially for languages that have a scarcity of available materials.

Therefore, the encoders in the M2 must encode the input in a way that enables any decoder to generate captions. The decoders of the M2 model must have the ability to generate output using the encoded information from any M2 encoder, demonstrating the model's versatility in handling different language combinations. By examining extensive language models, as exemplified in the research conducted by Brown et.al. (2020) [22], it becomes clear that larger models intrinsically contain the ability to perform very well in cases where only a little amount of training data is available. This phenomena highlights the significance of the size of the model in adapting to the necessary flexibility for successful few-shot learning tasks.

Qualitative Analysis

In the qualitative analysis of our few-shot image captioning model, we first examined the generated captions for their relevance and coherence. The model consistently demonstrated a strong ability to provide captions that were contextually relevant to the given images, showcasing a nuanced understanding of visual content. Analyzing the contents of Fig. 5.15, wherein the model undergoes training on the complete dataset, it becomes evident that the model excels in the generation of captions that precisely identify the child is skiing on a snow-covered slope within the depicted image. In Fig. 5.16, where

	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
Full dataset	74.50	66.11	58.76	52.07	62.86	56.44	1.2043
Few-shot: 2000 Samples	74.50	66.11	58.76	52.07	62.86	58.40	1.2616
Few-shot: 1000 Samples	71.90	62.84	55.24	48.43	59.60	56.37	1.129
Few-shot: 500 Samples	67.24	57.03	49.02	42.09	53.84	52.16	0.94
Zero-shot	34.51	17.18	9.09	5.09	16.46	24.34	0.1228

Table 5.11: Performance Metrics comparing the Few-shot perform on various sample size using CNN encoder-Marian NMT decoder: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.

	BL1	BL2	BL3	BL4	Avg.BL	R-L	CDr
Full dataset	80.3	72.38	65.41	59.05	69.39	63.25	1.6421
Few-shot: 2000 Samples	72.42	62.41	54.20	47.08	59.02	55.43	1.1886
Few-shot: 1000 Samples	70.25	60.37	52.08	44.92	56.90	54.33	1.0880
Few-shot: 500 Samples	65.01	54.08	45.71	38.55	50.83	49.63	0.8730

Table 5.12: Performance Metrics comparing the Few-shot perform on various sample size using CNN encoder- modified Marian NMT decoder: A analysis of BL1: BLEU-1, BL2: BLEU-2, BL3: BLEU-3, BL4: BLEU-4, Avg.BL: Average BLEU Scores, R-L: ROUGE-L, and CDr: CIDEr scores across various approaches.

the dataset is constrained to 2000 samples, there is a marginal decline in the model's performance compared to the full dataset model. Despite this reduction, the model remains adept at delivering reasonably accurate results, correctly discerning the presence of a small child in the snowy scene. Further reducing the sample size to 1000, as depicted in Fig. 5.17, results in a notable decline in accuracy. In contrast to previous identifications of a child, the model now recognizes a man, stating "a man on skis on a hillside." This deviation suggests a sensitivity to reduced data, leading to variations in the model's captioning outputs. Continuing the reduction in sample size to 500, as depicted in Fig. 5.18, there is a discernible but subtle decrease in accuracy. While the model still identifies a child participating in skiing, it introduces a broader interpretation, describing the scene as involving a couple of people in the snowy environment.

Even with diminishing sample sizes, the few-shot setup continues to showcase its effectiveness in generating fluent captions across various scenarios. Although there is a gradual reduction in accuracy, the models consistently exhibit an impressive ability to capture the essence of the depicted image. This persistence underscores the robustness of the few-shot learning approach in effectively handling diverse datasets.

5.3.6 Research Question: 06

How do the results of image captioning using original data compare to those obtained through machine-translated data?

Quantitative Analysis

In Table 5.13, the initial row presents the outcomes of the translation training experiment. Here, we translated the original English COCO dataset to Italian using a machine translation model. To maintain consistency in the neural machine translation (NMT)

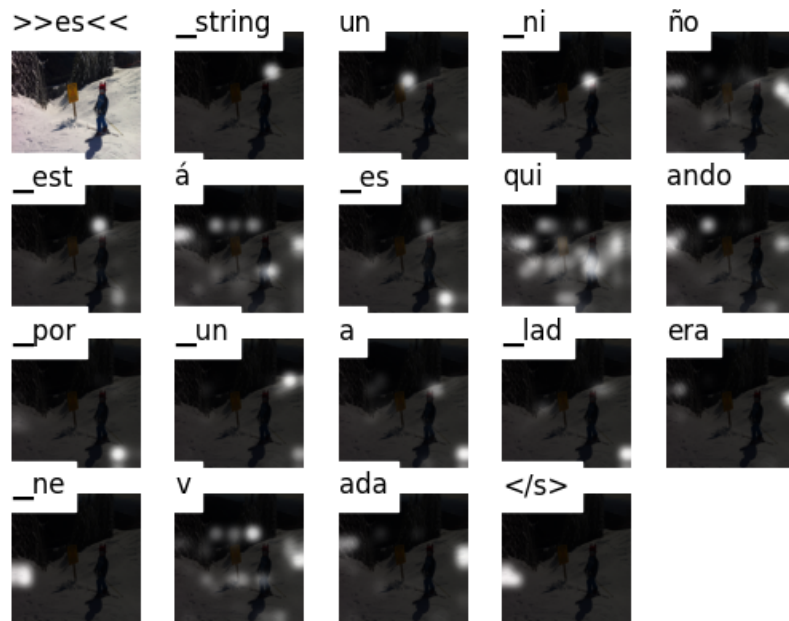


Figure 5.15: Analyzing CNN-modified Pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

model, we utilized the Marian NMT model for translating the English captioning dataset into Italian. Subsequently, we trained our image captioning model with this translated Italian dataset. The results displayed indicate a notable improvement when compared to our single model in Italian language. Both these model achieve a BLEU-4 score of 48. However, it is crucial to acknowledge that this model encounters its own limitations, particularly in terms of the fluency of generated captions. In the translate-test experiment, our approach involved initially training a captioning model using the English COCO dataset. Subsequently, we leveraged the inference generated from this captioning data to perform translations into the target language, which, in this case, was Italian. To assess the accuracy of the translated Italian inference captions, we conducted a comparison using the BLEU score between the translated test data in Italian and the original Italian captioning test data. This evaluation provided insights into the fidelity and precision of the Italian captions produced through the translation process. It is evident that the results exhibit poor performance, indicated by a BLEU-1 score of 14.

Qualitative Analysis

Upon analysis as illustrated in Fig5.19, it becomes apparent that the translation-trained model renders the caption as "a black cat lying down." However, it fails to capture certain visual details. In contrast as shown in Fig.5.20, the image captioning model, not only conveys the cat's posture but also extracts the additional information that the cat is

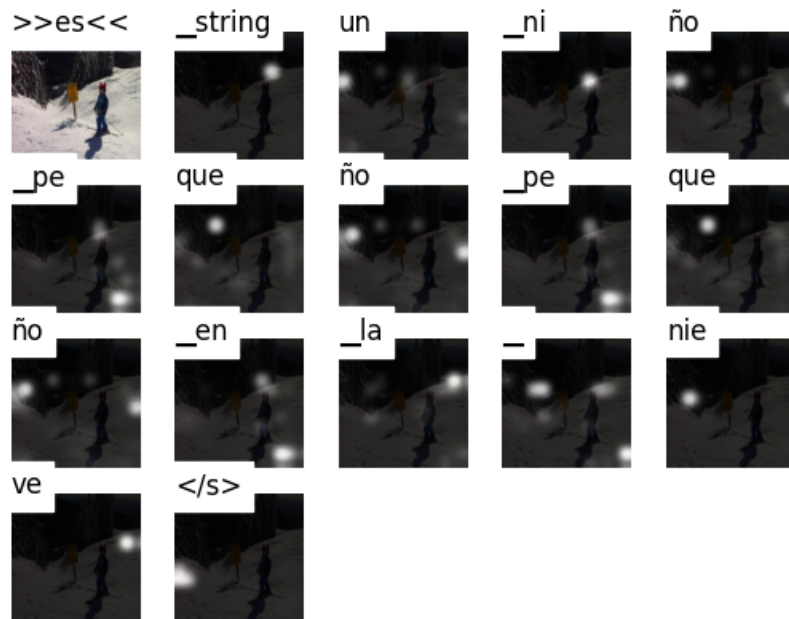


Figure 5.16: Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

lying on grass. This comparison underscores the potential limitations of relying on a translation model, especially when minute visual details are crucial.

Utilizing a translation model to convert captions into a target language gives rise to issues regarding the potential diminishment of complex information. Relying exclusively on English captions as an intermediate creates a bottleneck in the process. Using a translation model to translate faulty English captions may unintentionally spread inaccuracies to the translated annotations in the target language. In order to tackle these issues, there is a pressing need for multi-modal systems that effectively combine visual and textual information in a seamless manner. Implementing such technologies would guarantee a more thorough depiction, reducing the likelihood of data loss and errors while generating captions. This approach can significantly enhance the fidelity of cross-modal tasks, offering a more reliable and accurate portrayal of visual content in different languages.

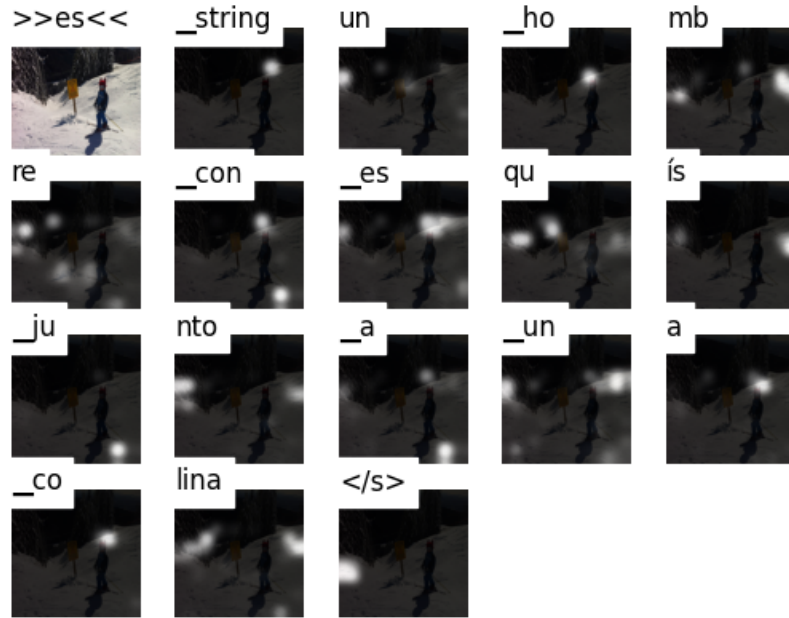


Figure 5.17: Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model’s inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

	BL1	BL2	BL3	BL4
translate-train	72.68	62.42	54.21	48.35
Translate-test	14.11	1.78	-	-

Table 5.13: Performance Metrics for Italian caption Generation using translate-train (translate the English data to Italian using NMT model) and translate-test methods (translate English inference captions to Italian using NMT model): A Comparative Analysis of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and CIDEr Scores Across Various Models

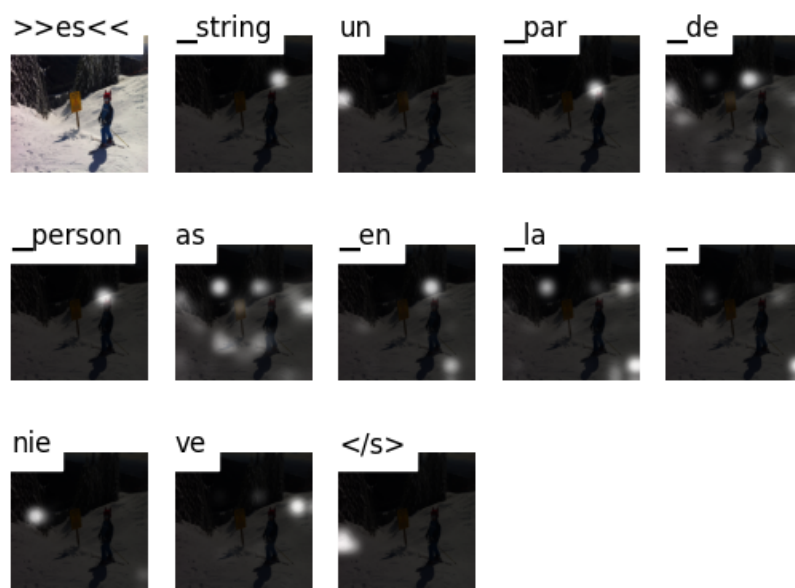


Figure 5.18: Analyzing CNN encoder-modified pretrained MarianNMT transformer for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

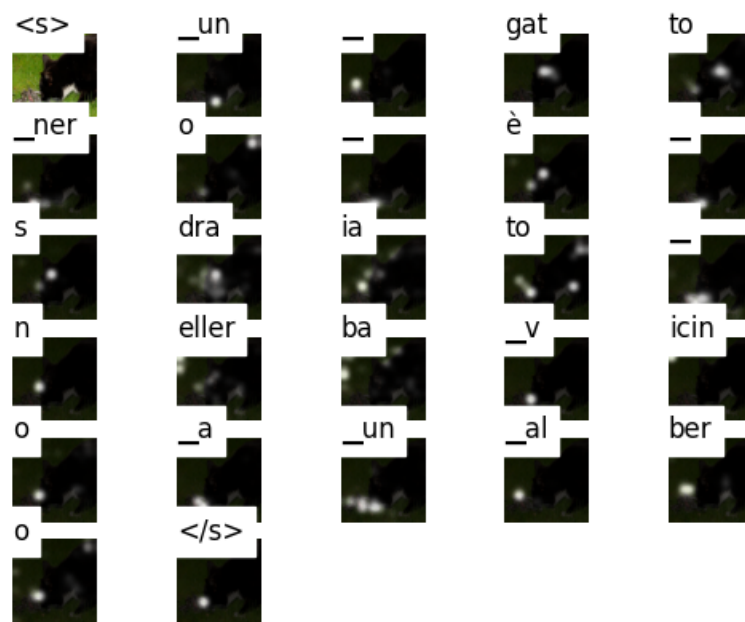


Figure 5.19: Analyzing CNN encoder-pretrained Marian NMT transformer decoder for Single model using Italian translation of English Dataset: Saliency map depicting the model's inference on translated dataset, offering a visual representation of attention and focus areas in caption generation

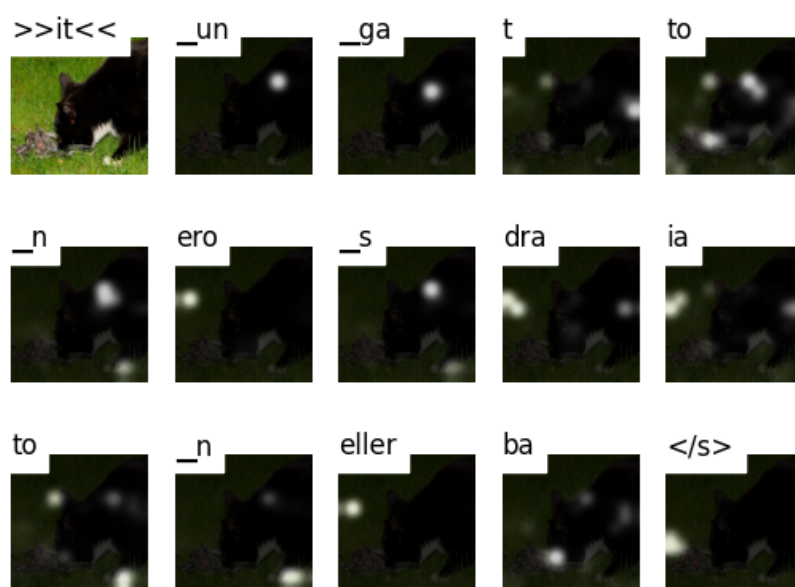


Figure 5.20: Analyzing CNN encoder-pretrained Marian NMT transformer decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

Chapter 6

Conclusion

In the concluding chapter, we bring together the threads of our exploration, extracting valuable insights from the thorough examination and results outlined in this report. Next, we examine the limitations of our approach and outline the future prospects of the research.

6.1 Conclusion

In conclusion, the task of image captioning poses a formidable challenge for machine intelligence, weaving together intricacies from both Computer Vision and Natural Language Processing domains. Furthermore, there is no clear definition for the captioning task itself, and there are numerous ways to generate captions with varied goals and styles. Notably, a significant limitation lies in the predominantly English-centric exploration of image captioning, excluding non-English speakers from the benefits of technologies. Recognizing this void, researchers are now actively exploring into the realm of cross-lingual image captioning, seeking to bridge the language gap and make these advancements accessible on a global scale.

In summary, this report presents a comprehensive evaluation and adaptation of the M2 model initially proposed by Lyu et al [57] for multilingual machine translation. By redesign the M2 architecture for the image captioning task. We conduct a systematic exploration of various decoder models, including LSTM, Transformer, and pre-trained transformer. Our findings unequivocally point to the pre-trained transformer as the most effective choice. Furthermore, through additional modifying the pre-trained transformer with 1x1 convolution layer, we ultimately found that the adapted transformer outperforms the alternative models.

Through extensive comparisons across single models, 1-1 models, and the M2 model under diverse conditions, our investigation reveals the substantial benefits of multi-way training for the M2 architecture. Notably, our research extends beyond conventional settings to demonstrate the adaptability of the M2 in low-resource scenarios, showcasing its capacity to generate meaningful captions even in a few-shot setting. The M2 has proven to be versatile and adaptable to a wide range of training environments, making it

an effective choice for image captioning for multi-lingual contexts. This will especially help low-resource languages in few-shot settings by cross-lingual transfer learning, transferring knowledge from high-resource languages.

In conclusion, our study not only validates the suitability of the M2 model for image captioning but also underscores the significance of leveraging pre-trained NMT transformer architectures. The demonstrated versatility of the M2, coupled with its capacity to thrive in varied training conditions, positions it as a robust choice for advancing the capabilities of image captioning technologies across multilingual and resource-constrained contexts.

6.2 Limitations

In this section, we will draw attention to a few limitations of our proposed approach:

1. Our study is limited by its sole reliance on the MS-COCO dataset. While this is a novel benchmark, other datasets like Flickr30K [63] or Conceptual12M [23] may introduce unique challenges, such as diverse image characteristics and linguistic complexities, that our architecture has not been tested on.
2. In our research, we used the Marian Neural Machine Translation (NMT) framework from Hugging Face. Using a different pre-trained NMT transformer may present different technical difficulties. Implementing another pre-trained model would necessitate significant changes to ensure alignment between the image captioning encoder and the characteristics of the NMT decoder model.
3. Choosing a different NMT model, other than MarianNMT, could lead to variations in the quality of captions. Different models can demonstrate a range of linguistic capabilities, which can impact the accuracy and dependability of the produced captions.
4. Our analysis uncovered random linguistic errors that occur while employing a 1-1 model with language tokens superimposed on the image, which impacts the overall performance of the model. Therefore, it is necessary to improve the process of cross-modal grounding.

6.3 Future works

In this section, we outline potential directions for the future of this research:

1. Conduct experiments on diverse datasets like Flickr30K [63] or Conceptual12M [23] with the aim of improving the models' ability to generalize and withstand variations. Through the evaluation of the model on diverse datasets, it is interesting to determine its ability to perform in a broad range of scenarios.
2. Improve the model's adaptability by introducing a more modular framework. This involves organizing each component of the model into distinct and interchangeable modules, facilitating simpler modification and adaptation. A modular design allows for the smooth integration of new language in multilingual captioning system.

3. Improving cross-modal grounding to enhance the interaction between different modalities in a more effective and detailed manner [87].
4. Optimize the performance of the M2 architecture specifically for zero-shot scenarios.
5. Investigating various fusion mechanisms holds great potential as an interesting path for future research. The objective is to expand the limits of multi-modal learning by improving the interaction between the Image Encoder and Text Decoder. Different fusion mechanisms can include attention mechanisms [76], cross-modal interactions [76], or novel structures that effectively integrate visual and textual information.
6. An interesting approach involves substituting the current self/cross attention processes with other linear attention strategies, such as Efficient Attention [68] and Flash Attention [26]. Exploring the integration of these attention mechanisms into our framework has the potential to enhance information processing by making it more efficient and effective.
7. Replacing the existing pretrained transformer text decoder by including advanced causal Language Models (LLMs) such as Llama 2 [74], Mistral 13B [41], and OPT [86].

Bibliography

- [1] 2024. The architecture of Convolutional Neural Networks(CNN). <https://www.analyticssteps.com/blogs/convolutional-neural-network-cnn-graphical-visualization-code-explanation> (2024). Accessed on January 28, 2024.
- [2] 2024. A Battle Against Amnesia: A Brief History and Introduction of Recurrent Neural Networks. <https://towardsdatascience.com/a-battle-against-amnesia-a-brief-history-and-introduction-of-recurrent-neural-> (2024). Accessed on January 28, 2024.
- [3] 2024. Convolutional Neural Networks for Text Classification. <https://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/> (2024). Accessed on January 28, 2024.
- [4] 2024a. Ethnologue: How many languages are there in the world? <https://www.ethnologue.com/guides/how-many-languages> (2024). Accessed on January 28, 2024.
- [5] 2024. García C., MS-COCO-ES, (2020), GitHub repository. <https://github.com/carlosGarciaHe/MS-COCO-ES> (2024). Accessed on January 28, 2024.
- [6] 2024. HDF5 for Python. <https://github.com/Mimino666/langdetect> (2024). Accessed on January 28, 2024.
- [7] 2024. Image captioning example. https://evergreen.team/assets/images/articles/machine-learning/image_captioning_train.png (2024). Accessed on January 28, 2024.
- [8] 2024b. langdetect. <https://www.h5py.org/> (2024). Accessed on January 28, 2024.
- [9] 2024. MarianNMT. https://huggingface.co/docs/transformers/model_doc/marian (2024). Accessed on January 28, 2024.
- [10] 2024. Natural Language Toolkit (NLTK). <https://github.com/nltk/nltk> (2024). Accessed on January 28, 2024.
- [11] 2024. Residual blocks — Building blocks of ResNet. <https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec> (2024). Accessed on January 28, 2024.
- [12] 2024. RESNET101. <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet101.html> (2024). Accessed on January 28, 2024.

- [13] 2024. The Transformer Model. <https://machinelearningmastery.com/the-transformer-model/> (2024). Accessed on January 28, 2024.
- [14] 2024. Understanding LSTM Networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (2024). Accessed on January 28, 2024.
- [15] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* (2018).
- [16] Aliko Anagnostopoulou, Mareike Hartmann, and Daniel Sonntag. 2023. Putting Humans in the Image Captioning Loop. *arXiv preprint arXiv:2306.03476* (2023).
- [17] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* 29 (2016).
- [18] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1409.0473>
- [19] Rajarshi Biswas, Michael Barz, Mareike Hartmann, and Daniel Sonntag. 2021. Improving German Image Captions Using Machine Translation and Transfer Learning. In *Statistical Language and Speech Processing - 9th International Conference, SLSP 2021, Cardiff, UK, November 23-25, 2021, Proceedings (Lecture Notes in Computer Science)*, Luis Espinosa Anke, Carlos Martín-Vide, and Irena Spasic (Eds.), Vol. 13062. Springer, 3–14. DOI:http://dx.doi.org/10.1007/978-3-030-89579-2_1
- [20] Rajarshi Biswas, Michael Barz, and Daniel Sonntag. 2020. Towards explanatory interactive image captioning using top-down and bottom-up features, beam search and re-ranking. *KI-Künstliche Intelligenz* 34 (2020), 571–584.
- [21] Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 111–118.
- [22] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>

- [23] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 3558–3568. DOI:
<http://dx.doi.org/10.1109/CVPR46437.2021.00356>
- [24] Aozhu Chen, Xinyi Huang, Hailan Lin, and Xirong Li. 2020. Towards annotation-free evaluation of cross-lingual image captioning. In *MMAsia 2020: ACM Multimedia Asia, Virtual Event / Singapore, 7-9 March, 2021*, Tat-Seng Chua, Jingdong Wang, Qi Tian, Cathal Gurrin, Jia Jia, Hanwang Zhang, and Qianru Sun (Eds.). ACM, 69:1–69:7. DOI:
<http://dx.doi.org/10.1145/3444685.3446322>
- [25] Xianyu Chen, Ming Jiang, and Qi Zhao. 2021. Self-distillation for few-shot image captioning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 545–555.
- [26] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html
- [27] Xuanyi Dong, Linchao Zhu, De Zhang, Yi Yang, and Fei Wu. 2018. Fast parameter adaptation for few-shot image captioning and visual question answering. In *Proceedings of the 26th ACM international conference on Multimedia*. 54–62.
- [28] Desmond Elliott, Stella Frank, and Eva Hasler. 2015. Multilingual image description with neural sequence models. *arXiv preprint arXiv:1510.04709* (2015).
- [29] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German Image Descriptions. In *Proceedings of the 5th Workshop on Vision and Language, hosted by the 54th Annual Meeting of the Association for Computational Linguistics, VL@ACL 2016, August 12, Berlin, Germany*. The Association for Computer Linguistics. DOI:
<http://dx.doi.org/10.18653/V1/W16-3210>
- [30] Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, 944–948. DOI:
<http://dx.doi.org/10.18653/V1/2021.EACL-MAIN.80>
- [31] Jiahui Gao, Yi Zhou, Philip L. H. Yu, Shafiq R. Joty, and Jiuxiang Gu. 2022. UNISON: Unpaired Cross-Lingual Image Captioning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 10654–10662. DOI:
<http://dx.doi.org/10.1609/AAAI.V36I10.21310>
- [32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

- [33] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610. DOI:<http://dx.doi.org/10.1016/j.neunet.2005.06.042>
- [34] Jiuxiang Gu, Shafiq R. Joty, Jianfei Cai, and Gang Wang. 2018. Unpaired Image Captioning by Language Pivoting. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I (Lecture Notes in Computer Science)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.), Vol. 11205. Springer, 519–535. DOI:http://dx.doi.org/10.1007/978-3-030-01246-5_31
- [35] Fredrik K. Gustafsson. 2017. Neural Image Captioning for Intelligent Vehicle-to-Passenger Communication. <https://api.semanticscholar.org/CorpusID:35378079>
- [36] Mareike Hartmann, Alik Anagnostopoulou, and Daniel Sonntag. 2022. Interactive machine learning for image captioning. *arXiv preprint arXiv:2202.13623* (2022).
- [37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [38] Sepp Hochreiter. 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* 6, 2 (1998), 107–116. DOI:<http://dx.doi.org/10.1142/S0218488598000094>
- [39] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [40] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research)*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 4904–4916. <http://proceedings.mlr.press/v139/jia21b.html>
- [41] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and others. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).
- [42] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 4565–4574. DOI:<http://dx.doi.org/10.1109/CVPR.2016.494>
- [43] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, and others. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* 5 (2017), 339–351.

- [44] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, Fei Liu and Thamar Solorio (Eds.). Association for Computational Linguistics, 116–121. DOI:<http://dx.doi.org/10.18653/V1/P18-4020>
- [45] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 3128–3137. DOI:<http://dx.doi.org/10.1109/CVPR.2015.7298932>
- [46] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [47] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014a. Multimodal Neural Language Models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014 (JMLR Workshop and Conference Proceedings)*, Vol. 32. JMLR.org, 595–603. <http://proceedings.mlr.press/v32/kiros14.html>
- [48] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014b. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539* (2014).
- [49] Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.
- [50] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. 127–133.
- [51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 1106–1114. <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- [52] Weiyu Lan, Xirong Li, and Jianfeng Dong. 2017. Fluency-guided cross-lingual image captioning. In *Proceedings of the 25th ACM international conference on Multimedia*. 1549–1557.
- [53] Chin-Yew Lin and Eduard H. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*, Marti A. Hearst and Mari Ostendorf (Eds.). The Association for Computational Linguistics. <https://aclanthology.org/N03-1020/>

- [54] Chin-Yew Lin and Franz Josef Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, Donia Scott, Walter Daelemans, and Marilyn A. Walker (Eds.). ACL, 605–612. DOI:<http://dx.doi.org/10.3115/1218955.1219032>
- [55] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V (Lecture Notes in Computer Science)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.), Vol. 8693. Springer, 740–755. DOI:http://dx.doi.org/10.1007/978-3-319-10602-1_48
- [56] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task Sequence to Sequence Learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1511.06114>
- [57] Sungwon Lyu, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 5905–5918. DOI:<http://dx.doi.org/10.18653/V1/2020.EMNLP-MAIN.476>
- [58] Burak Makav and Volkan Kılıç. 2019. A new image captioning approach for visually impaired people. In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE, 945–949.
- [59] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2015. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6632>
- [60] Sidharth Mehra. 2024. Relationship between AI, ML, DL and NLP. https://www.researchgate.net/figure/Relationship-between-AI-ML-DL-and-NLP-7_fig8_343079524 (2024). Accessed on January 28, 2024.
- [61] Takashi Miyazaki and Nobuyuki Shimizu. 2016. Cross-Lingual Image Caption Generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics. DOI:<http://dx.doi.org/10.18653/V1/P16-1168>
- [62] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 311–318. DOI:<http://dx.doi.org/10.3115/1073083.1073135>

- [63] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2641–2649. DOI:<http://dx.doi.org/10.1109/ICCV.2015.303>
- [64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research)*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 8748–8763. <http://proceedings.mlr.press/v139/radford21a.html>
- [65] Frank Rosenblatt and others. 1962. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Vol. 55. Spartan books Washington, DC.
- [66] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and others. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.
- [67] Antonio Scaiella, Danilo Croce, and Roberto Basili. 2019. Large scale datasets for Image and Video Captioning in Italian. *Italian Journal of Computational Linguistics* 2, 5 (2019), 49–60. http://www.ai-ic.it/IJCoL/v5n2/IJCOL_5_2_3___scaiella_et_al.pdf
- [68] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. 2021. Efficient Attention: Attention with Linear Complexities. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, 3530–3538. DOI:<http://dx.doi.org/10.1109/WACV48630.2021.00357>
- [69] NN Shinde, N Gawde, and N Paradkar. 2020. Social media image caption generation using deep learning. *International Journal of Engineering Development and Research* 8 (2020), 222–228.
- [70] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [71] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2023. From Show to Tell: A Survey on Deep Learning-Based Image Captioning. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 1 (2023), 539–559. DOI:<http://dx.doi.org/10.1109/TPAMI.2022.3148210>
- [72] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

- [73] Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT - Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisboa, Portugal, November 3-5, 2020*, Mikel L. Forcada, André F. T. Martins, Helena Moniz, Marco Turchi, Arianna Bisazza, Joss Moorkens, Ana Guerberof Arenas, Mary Nurminen, Lena Marg, Sara Fumega, Bruno Martins, Fernando Batista, Luísa Coheur, Carla Parra Escartín, and Isabel Trancoso (Eds.). European Association for Machine Translation, 479–480. <https://aclanthology.org/2020.eamt-1.61/>
- [74] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [75] Satoshi Tsutsui and David Crandall. 2017. Using artificial tokens to control languages for multilingual image caption generation. *arXiv preprint arXiv:1706.06275* (2017).
- [76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [77] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 4566–4575. DOI:<http://dx.doi.org/10.1109/CVPR.2015.7299087>
- [78] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 3156–3164. DOI:<http://dx.doi.org/10.1109/CVPR.2015.7298935>
- [79] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.
- [80] Lilian Weng. 2024. Multi-Head Attention. <https://paperswithcode.com/method/multi-head-attention> (2024). Accessed on January 28, 2024.
- [81] Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.
- [82] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, and others. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [83] Yike Wu, Shiwan Zhao, Jia Chen, Ying Zhang, Xiaojie Yuan, and Zhong Su. 2019. Improving Captioning for Low-Resource Languages by Cycle Consistency. In *IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019*. IEEE, 362–367. DOI:<http://dx.doi.org/10.1109/ICME.2019.00070>

- [84] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings)*, Francis R. Bach and David M. Blei (Eds.), Vol. 37. JMLR.org, 2048–2057. <http://proceedings.mlr.press/v37/xuc15.html>
- [85] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2020. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. <https://openreview.net/forum?id=BJgnXpVYwS>
- [86] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).
- [87] Xi Zhu, Zhendong Mao, Chunxiao Liu, Peng Zhang, Bin Wang, and Yongdong Zhang. 2020. Overcoming Language Priors with Self-supervised Learning for Visual Question Answering. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 1083–1089. DOI:<http://dx.doi.org/10.24963/IJCAI.2020/151>

Appendix A

Appendix

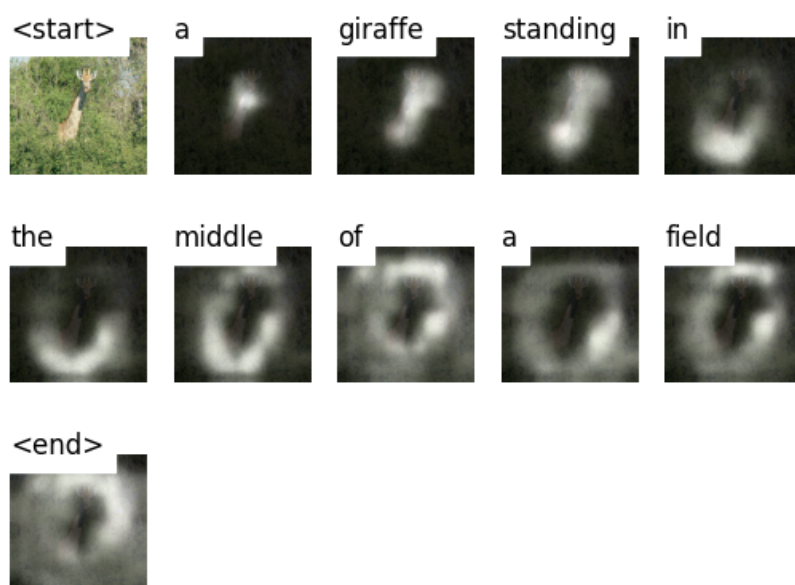


Figure A.1: Analyzing CNN encoder-LSTM decoder for Single model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

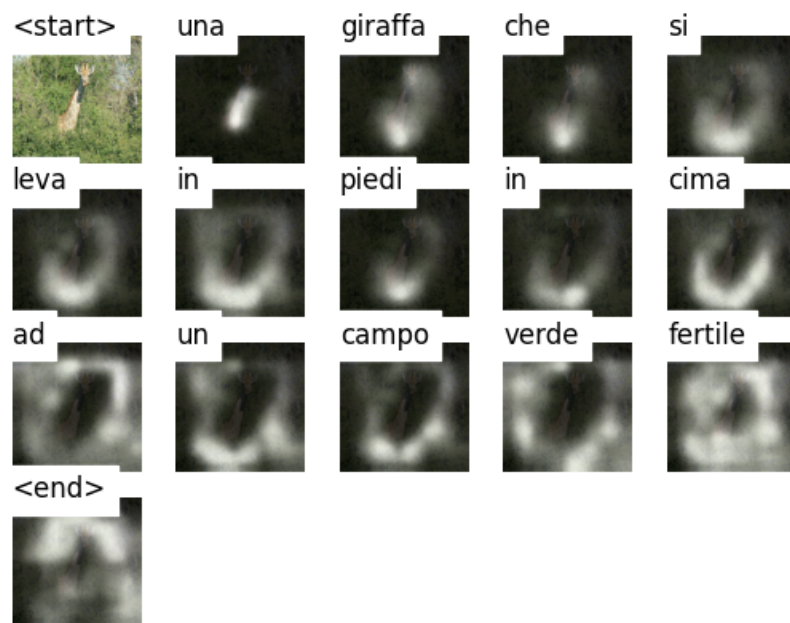


Figure A.2: Analyzing CNN encoder-LSTM decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.3: Saliency map of a CNN-TransformAnalyzing CNN encoder-transformer decoder for Single model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

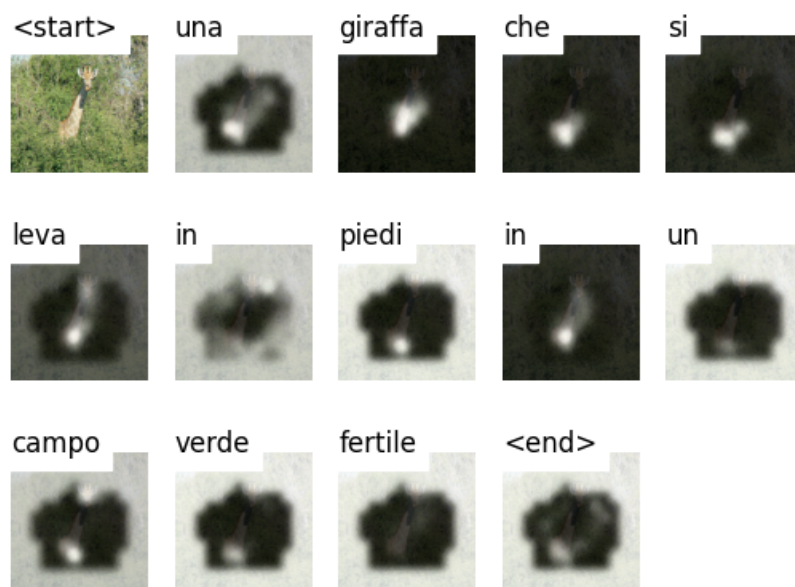


Figure A.4: Analyzing CNN encoder-transformer decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

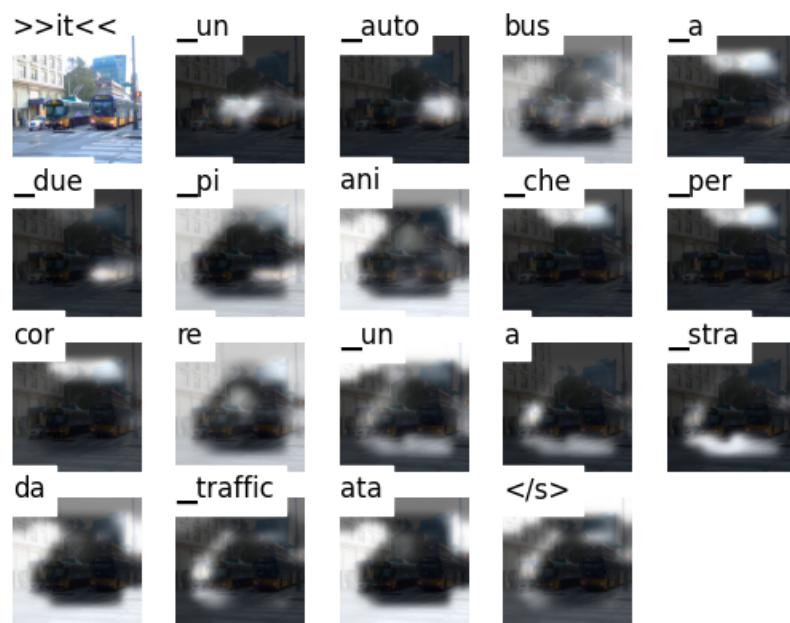


Figure A.5: Analyzing CNN encode-modified pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

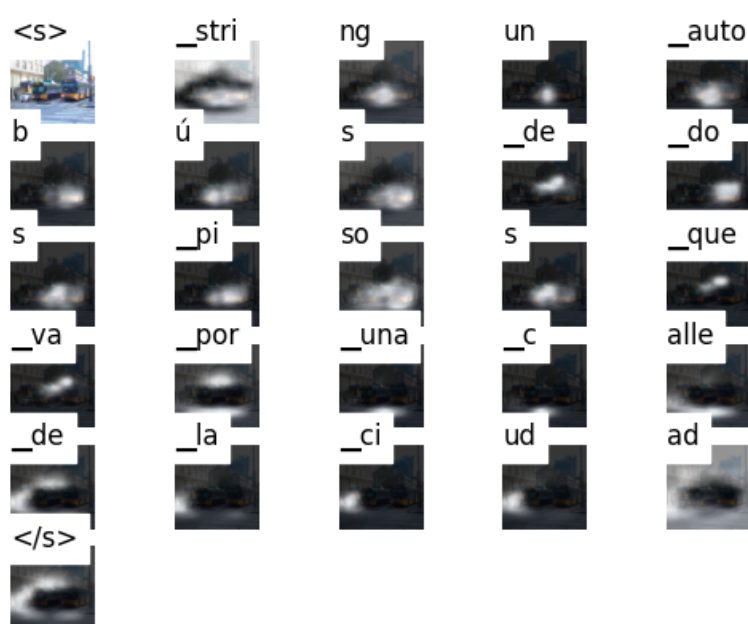


Figure A.6: Analyzing CNN encoder-modified pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

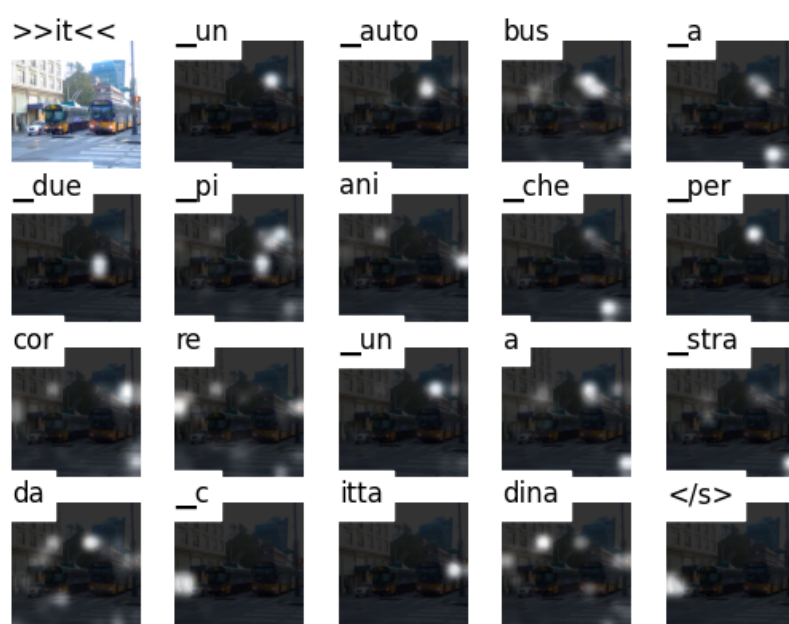


Figure A.7: Analyzing CNN encoder-pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.8: Analyzing CNN encoder-pretrained MarianNMT decoder for Single model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

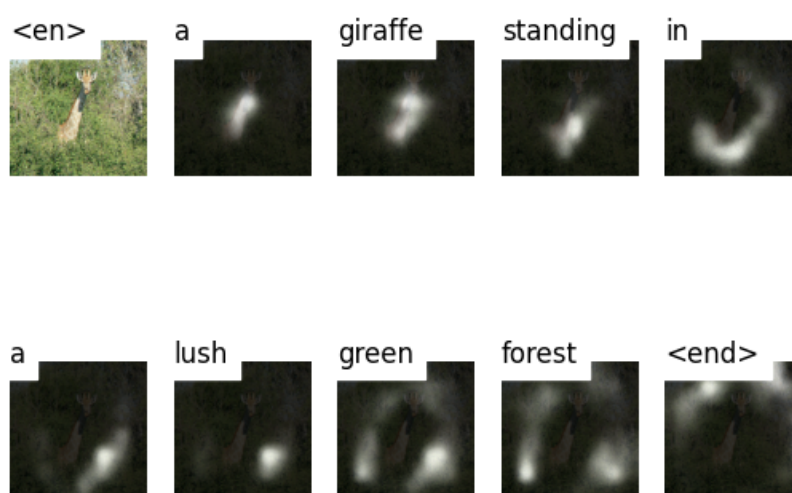


Figure A.9: Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.10: Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.11: Analyzing CNN encoder-LSTM decoder for 1-1 model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

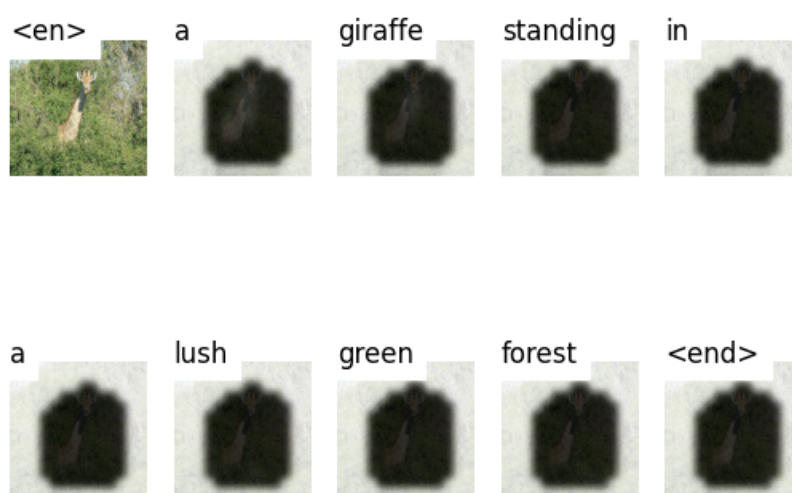


Figure A.12: Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on English MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.13: Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on Italian MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.14: Analyzing CNN encoder-transformer decoder for 1-1 model: Saliency map depicting the model's inference on Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

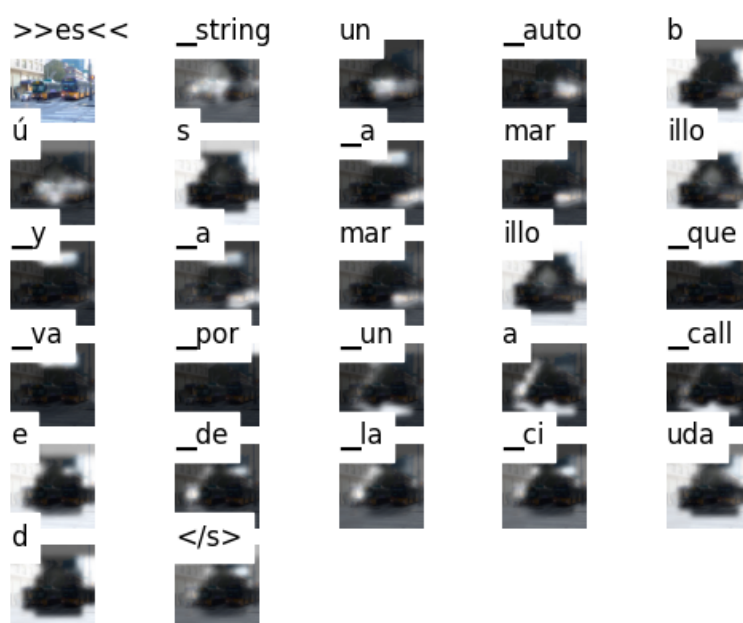


Figure A.15: Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

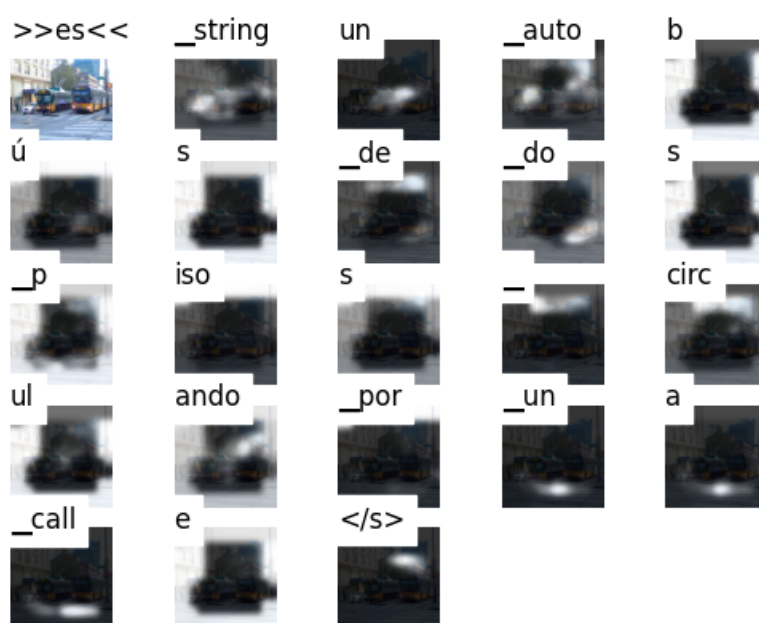


Figure A.16: Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

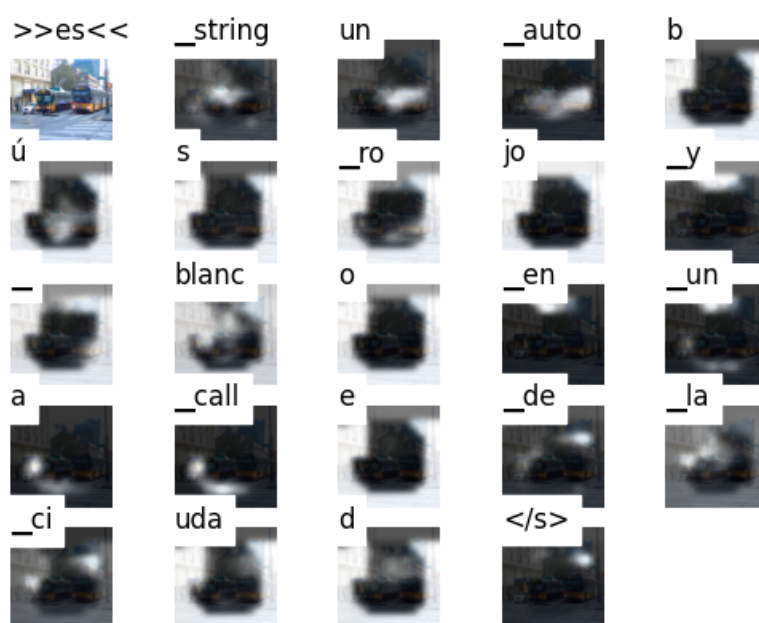


Figure A.17: Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation



Figure A.18: Analyzing CNN encoder-modified pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

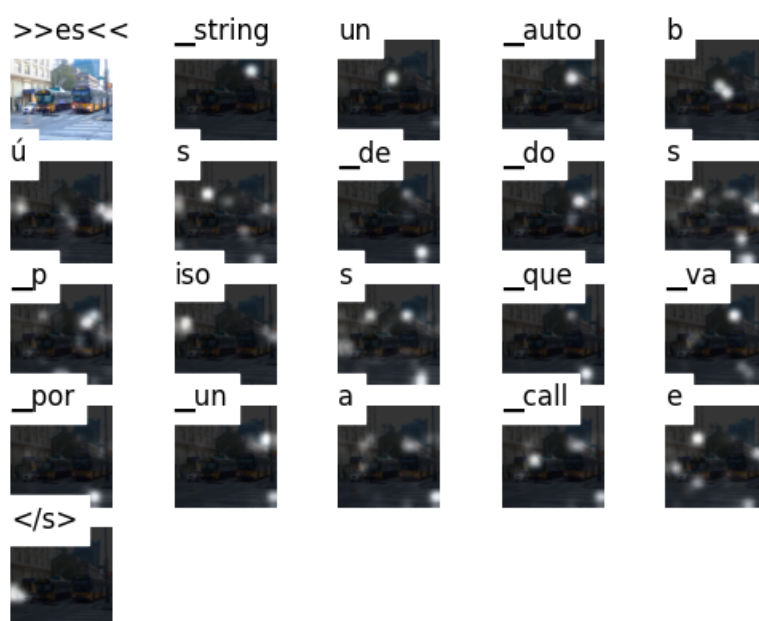


Figure A.19: Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on full Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

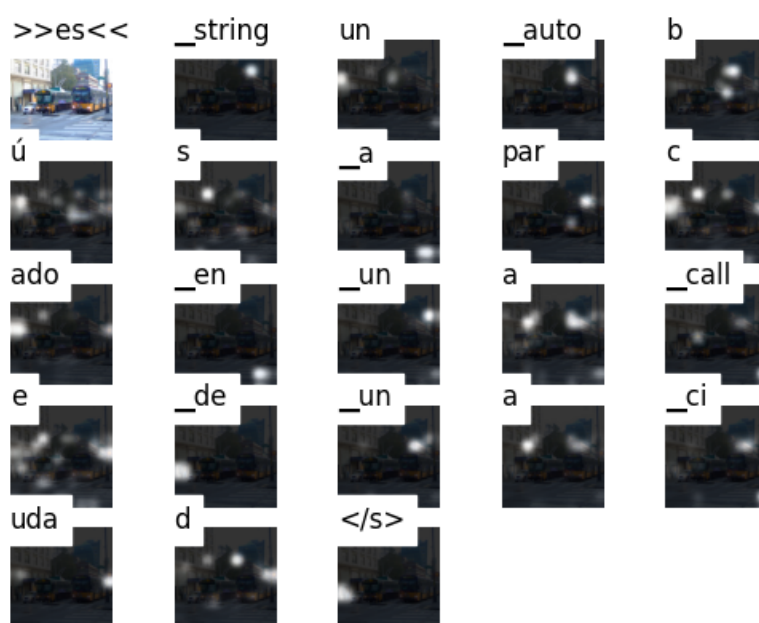


Figure A.20: Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 2000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

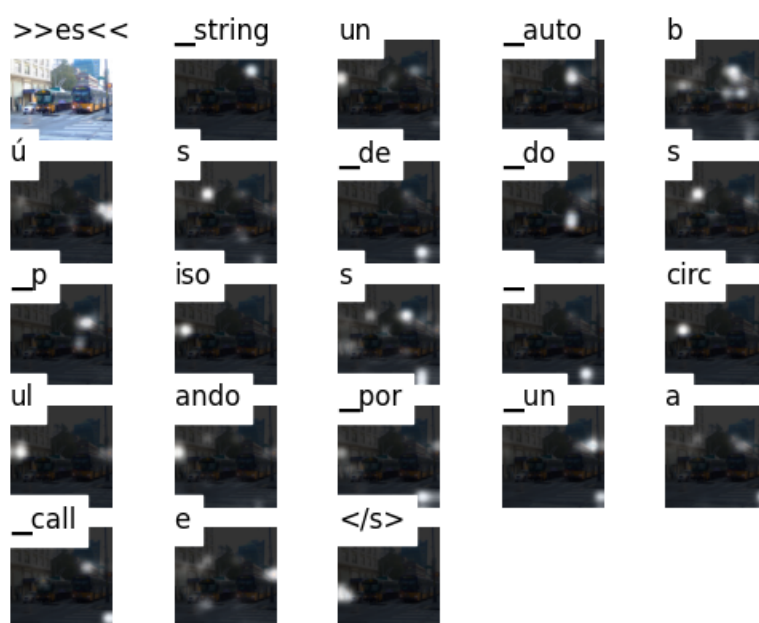


Figure A.21: Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 1000 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation

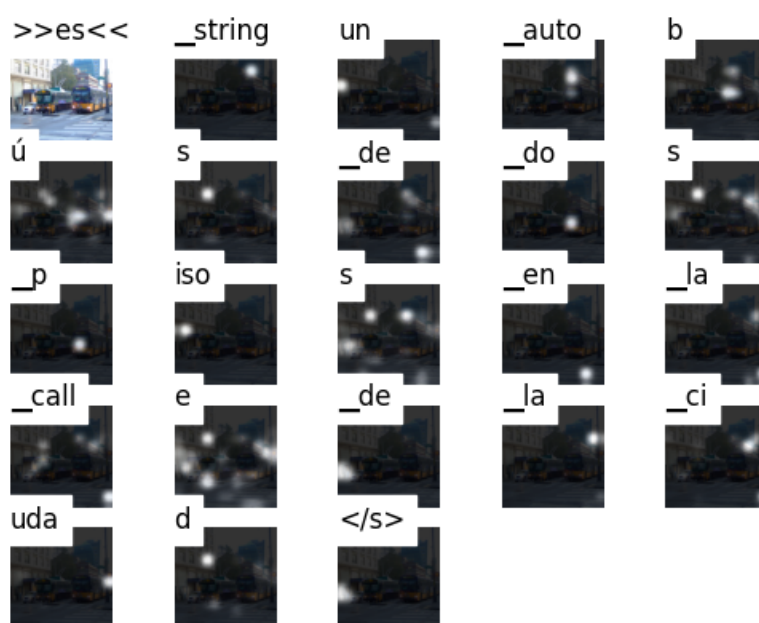


Figure A.22: Analyzing CNN encoder-pretrained MarianNMT decoder for M2 architecture in few-shot setting: Saliency map depicting the model's inference on 500 samples of Spanish MS-COCO dataset, offering a visual representation of attention and focus areas in caption generation