

FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND INDUSTRIAL
INFORMATICS

OLIVER CABRAL JORGE

CONTENT-BASED VIDEO RETRIEVAL FROM NATURAL
LANGUAGE

DISSERTATION

CURITIBA

2022

OLIVER CABRAL JORGE

**CONTENT-BASED VIDEO RETRIEVAL FROM NATURAL
LANGUAGE**

**Recuperação de vídeos baseada em conteúdo a partir de linguagem
natural**

Dissertation presented to the Graduate Program in Electrical Engineering and Industrial Informatics of the Federal University of Technology - Paraná (UTFPR) as part of the fulfillment of the requirements for the title of Master in Electrical Engineering and Industrial Informatics.

Advisor: Prof. Dr. Heitor Silvério Lopes

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



OLIVER CABRAL JORGE

CONTENT-BASED VIDEO RETRIEVAL FROM NATURAL LANGUAGE

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Data de aprovação: 12 de Setembro de 2022

Dr. Andre Eugenio Lazzaretti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. David Menotti Gomes, Doutorado - Universidade Federal do Paraná (Ufpr)

Dr. Luiz Celso Gomes Junior, Doutorado - Universidade Tecnológica Federal do Paraná

Pedro Henrique Bugatti, - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 12/09/2022.

This work is dedicated to my wife, who supported and encouraged me throughout the years keeping me in the right path, and my father, who taught me the importance of studying and that I wish he could be here to see this achievement.

ACKNOWLEDGEMENTS

This work is not mine alone, but the result of several indirect people who contributed in many different forms. In here, I would like to thank and recognize those who have helped me.

First of all, my wife, who stayed with me during those difficult years, giving me the strength and courage needed to complete this chapter in my life.

Secondly, to my advisor, Heitor, who was not only an advisor, but also a friend, being patient and trusting in me even with all the “rocks” that showed up in the path.

Also, my brother, who is an example of how to be true to yourself and make the most out of it.

Finally, to my friend Hellen, who helped me to start my Master’s and who was always a source of encouragement.

If you can't fly then run, if you can't run then
walk, if you can't walk then crawl, but whatever
you do you have to keep moving forward.
(JUNIOR, Martin Luther King, 1956).

RESUMO

JORGE, Oliver Cabral. **Recuperação de vídeos baseada em conteúdo a partir de linguagem natural**. 2022. 87 f. Dissertation (Master em Electrical Engineering and Industrial Informatics) – Federal University of Technology - Paraná. Curitiba, 2022.

Cada vez mais os vídeos estão se tornando os meios mais comuns de comunicação, alavancadas pela popularização de aparelhos acessíveis de gravação de vídeos e pelas redes sociais como TikTok, Instagram e demais. As formas mais comuns de pesquisa de vídeos nestas redes sociais bem como nos portais de buscas, se baseiam em metadados vinculados aos vídeos por meio de palavras-chaves e classificações prévias. No entanto, buscas por palavras-chaves dependem de um conhecimento exato do que se deseja, e não necessariamente podem ser eficientes ao tentar encontrar um determinado vídeo a partir de uma descrição, superficial ou não, de uma determinada cena, podendo incorrer em resultados frustrantes da busca. O objetivo deste trabalho é encontrar um determinado vídeo dentro de uma lista de vídeos disponíveis a partir de uma descrição textual em linguagem natural baseado apenas no conteúdo de suas cenas, sem depender de metadados previamente catalogados. A partir de um dataset contendo vídeos com um número definido de descrições de suas cenas, foi modelada uma rede siamesa com função de perda tripla para identificar, em um hiperespaço, as similaridades entre duas modalidades diferentes, sendo uma delas as informações extraídas de um vídeo, e a outra as informações extraídas de um texto em linguagem natural. A arquitetura final do modelo, bem como os valores de seus parâmetros, foi definida baseada em testes que seguiram os melhores resultados obtidos. Devido ao fato de que os vídeos não são classificados em grupos ou classes e considerando que a função de perda tripla se baseia em um texto âncora e dois exemplos de vídeos, um positivo e um negativo, foi identificada uma dificuldade na seleção de exemplos falsos necessários para o treinamento da arquitetura. Desta forma, também foram testados métodos de escolha de exemplos de vídeos negativos para treinamento utilizando uma escolha aleatória e uma escolha direcionada, baseada nas distâncias das descrições disponíveis dos vídeos em fase de treinamento, sendo a primeira a mais eficiente. Ao final dos testes, foi alcançado um resultado com presença exata do vídeo buscado em 10,67% dos casos no top-1 e em 49,80% dos casos no top-10. Mais do que os resultados numéricos, foi feita uma análise qualitativa dos resultados. Desta análise, foi identificado que o modelo não se comporta de forma satisfatória para buscas em palavras atômicas, com melhores resultados em descrições mais complexas. Os bons resultados também estão principalmente relacionados ao uso de verbos e substantivos, e menos aos adjetivos e advérbios. Ainda, observou-se que os vídeos retornados possuem, de alguma forma, similaridades de cenas ou de tópicos com o texto procurado, indicando que a rede identificou o significado do texto procurado. De maneira geral, os resultados obtidos são promissores e encorajam a continuidade da pesquisa. Trabalhos futuros incluirão o uso de novos modelos de extração de informação de vídeos e de textos, bem como maior aprofundamento na escolha de exemplos negativos de vídeos para reforçar o treinamento.

Palavras-chave: Multimodalidade. Deep Learning. Recuperação de Vídeo. Linguagem Natural.

ABSTRACT

JORGE, Oliver Cabral. **Content-Based Video retrieval from natural language**. 2022. 87 p. Dissertation (Master's Degree in Electrical Engineering and Industrial Informatics) – Federal University of Technology - Paraná. Curitiba, 2022.

More and more, videos are becoming the most common means of communication, leveraged by the popularization of affordable video recording devices and social networks such as TikTok, Instagram, and others. The most common ways of searching for videos on these social networks as well as on search portals are based on metadata linked to videos through keywords and previous classifications. However, keyword searches depend on exact knowledge of what you want and may not necessarily be efficient when trying to find a particular video from a description, superficial or not, of a particular scene, which may lead to frustrating results in the search. The objective of this work is to find a particular video within a list of available videos from a textual description in natural language based only on the content of its scenes, without relying on previously cataloged metadata. From a dataset containing videos with a defined number of descriptions of their scenes, a Siamese network with a triplet loss function was modeled to identify, in hyperspace, the similarities between two different modalities, one of them being the information extracted from a video, and the other information extracted from a text in natural language. The final architecture of the model, as well as the values of its parameters, was defined based on tests that followed the best results obtained. Because videos are not classified into groups or classes and considering that the triplet loss function is based on an anchor text and two video examples, one positive and one negative, a difficulty was identified in the selection of false examples needed for the model training. In this way, methods of choosing examples of negative videos for training were also tested using a random choice and a directed choice, based on the distances of the available descriptions of the videos in the training phase, being the first the most effective. At the end of the tests, a result was achieved with the exact presence of the searched video in 10.67% of the cases in the top 1 and 49.80% of the cases in the top 10. More than the numerical results, a qualitative analysis of the results was conducted. From this analysis, it was identified that the model does not behave satisfactorily for searches in atomic words, with better results in more complex descriptions. Satisfactory results are also mainly related to the use of verbs and nouns, and less to adjectives and adverbs. Still, it was observed that the returned videos have, in some way, similarities of scenes or topics with the searched text, indicating that the network identified the meaning of the original text query. In general, the results obtained are promising and encourage the continuity of the research. Future work will include the use of new models for extracting information from videos and texts, as well as further studies into the controlled choice of negative video examples to reinforce training.

Keywords: Multimodality. Deep Learning. Video Retrieval. Natural Language.

LIST OF FIGURES

Figure 1 – McCulloch-Pitts Neuron representation. The neuron accepts binary values (X_n) that feeds an aggregation function (f) to formulate a binary output (g).	18
Figure 2 – Deep Learning (DL) representation. Layers of Neural Networks (NN) are connected one after the other increasing the number of calculated weights as it gets deeper.	19
Figure 3 – Convolutional Neural Network (CNN) representation. The input layer receives the data to be processed. The hidden layer performs convolution and pooling using a kernel to reduce the size of the original data. The output layer performs the classification.	20
Figure 4 – Recurrent Neural Network (RNN) representation. The connection between nodes creates a temporal internal state memory.	21
Figure 5 – Long Short-Term Memory (LSTM) representation of a cell with its internal Forget, Input and Output gates.	22
Figure 6 – Gated Recurrent Unit (GRU) representation of a cell with its internal Reset and Update gates.	22
Figure 7 – Global Vectors (GloVe) nearest neighbor results for the word “frog”.	25
Figure 8 – GloVe linear representation of semantic similarity.	25
Figure 9 – Bidirectional Encoder Representations from Transformers (BERT) phase representations. The pre-trained phase was trained in a large free text corpus, while the fine-tuning phase uses labeled data to fine-tune its learning capabilities, being tested in the major tasks: Stanford Question Answering Dataset (SQuAD), Multi-Genre Natural Language Inference (MNLI) and Named Entity Recognition (NER)	26
Figure 10 – Visual representation of the filtering steps.	30
Figure 11 – A visual representation of how the research areas was focused over the years.	31
Figure 12 – Top-5 most used datasets referenced per year.	41
Figure 13 – Solution architecture.	45
Figure 14 – Top 30 most common words per selected classes in the used dataset Corpus.	47
Figure 15 – Video feature extractor workflow. Frame-level features are extracted using CNN models which, in turn, are used as input to a RNN model, thus resulting in a spatial-temporal representation of the video.	48
Figure 16 – Video clip comparison to exemplify the difference in lengths and frame shapes.	48
Figure 17 – Number of maximum frames per video clip.	49
Figure 18 – Pair of feature and mask vectors. Each frame of a video has its extracted features filling a position of the vector. Parallel, a mask vector informs that the vector position is a valid one.	50
Figure 19 – Natural Language (NL) feature extraction workflow. Meaningful words from the sentence are converted to numeric representations that are used as input to a RNN model, resulting in a positional-temporal representation of the sentence.	51
Figure 20 – Siamese model showing the video and NL different input models sharing the same dense layers to produce similar results that can be compared in f_s Siamese function to calculate the Euclidean distance E_d	54

Figure 21 – Triplet-loss function illustration. Based on an anchor, a positive and a negative example, the model is trained to minimize the distance between positive examples and maximize it, otherwise.	54
Figure 22 – A NL sentence as an anchor, and a pair of videos as a positive and a negative example.	55
Figure 23 – Ranked negative example texts. The ranked sentences in the middle column are ranked as opposite to the anchor sentence in the left column. The related distance similarity found is listed in the right column.	57
Figure 24 – Example of the rules applied to select the negative examples. The steps taken to select four negative examples for each positive video are shown, from the starting video followed by three selection loops.	58
Figure 25 – Loss curve for comparing the optimizers.	62
Figure 26 – The triplet function architecture.	64
Figure 27 – Grid results explanation – the left square is the ground truth, while the squares number 1, 2, 3 and 4 represents the 1 st , 2 nd , 3 rd and 4 th ranked results. At the top of each square, the related NL sentence is shown.	70
Figure 28 – Retrieval example matching the 1 st ranked image.	71
Figure 29 – Retrieval example matching the 2 nd ranked image.	71
Figure 30 – Retrieval example matching the 3 rd ranked image.	72
Figure 31 – Retrieval example with no match on ranked image.	72
Figure 32 – Retrieval example for random free sentences.	75
Figure 33 – Retrieval example for video observation free sentences.	76

LIST OF TABLES

Table 1 – Summary of results of video retrieval works.	34
Table 2 – Top-5 most used datasets as identified in Section 2.2.2	41
Table 3 – Top 5 most used datasets comparison	44
Table 4 – Dataset statistics showing the total amount of available video clips per dataset, the total duration in minutes, the number of people described, and the number of available descriptions.	46
Table 5 – CNN models architecture comparison.	50
Table 6 – NLTK grammatical classes conversion used.	52
Table 7 – Top-k results in percentage for model architecture comparison.	61
Table 8 – Top-k results in percentage, for training with reduced layer’s size.	61
Table 9 – Top-k results in percentage for comparing the optimizers.	62
Table 10 – Top-k results in percentage for video dropout (d) variation.	63
Table 11 – Top-k results in percentage for Natural Language Processing (NLP) dropout (d) variation.	63
Table 12 – Top-k results in percentage for video feature extractor models.	65
Table 13 – Top-k results in percentage for training using different grammatical classes.	66
Table 14 – Top-k results in percentage for Triplet-loss impact analysis over α variation.	67
Table 15 – Top-k results in percentage for the impact of controlling the distance d of negative examples for a given anchor.	68
Table 16 – For negative examples for a given anchor, top-k yields a percentage of the balanced distribution of controlled distance d . The percentage distributions indicate the weights applied considering the left number as the most distant to the anchor and the right one as the closest.	69

LIST OF ACRONYMS

ACRONYMS

ActivityNet	Activity Net dataset
Adam	Adaptive Moment Estimation
AMT	Amazon Mechanical Turk
ANN	Artificial Neural Networks
API	Application Programming Interface
ASR	Automated Speech Recognition
AttNet	Attribute Detection Network
BERT	Bidirectional Encoder Representations from Transformers
BiGRU	Bidirectional Gated Recurrent Unit
BoW	Bag-of-Words
C3D	3D Convolutional Network
CCA	Cascade Cross-Modal Attention
CCL	Creative Commons License
Charades	Combining Human Assessment and Reasoning Aids for Decision-Making in Environmental Emergencies dataset
Charades-STA	Charades - Sentence Temporal Annotations dataset
CHD	Convolutional Hierarchical Decoder
CNN	Convolutional Neural Network
CV	Computer Vision
DAP	Deep Action Proposals
DiDeMo	Distinct Describable Moment dataset
DL	Deep Learning
FFMPEG	Fast Forward Moving Picture Experts Group
FPS	Frames per Second
GCN	Graph Convolutional Network
GloVe	Global Vectors
GNN	Graph Neural Network
GPU	Graphics Processing Units
GRU	Gated Recurrent Unit
HD	High Definition
IGAN	Iterative Graph Adjustment Network
InceptionV3	Inception-V3 Network
JST	Joint Semantic Tensor

LST	Latent Semantic Tree
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked LM
MNLI	Multi-Genre Natural Language Inference
MobileNetV2	MobileNet-V2 Network
MSR-VTT	MSR-Video to Text dataset
NER	Named Entity Recognition
NL	Natural Language
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Networks
NSP	Next Sentence Prediction
OCR	Optical Character Recognition
OpenCV	Open Source Computer Vision Library
PoS	Part-of-Speech
R@K	Recall at k
ResNet50	Residual Network - 50
RGB	Red, Green and Blue
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
sBert	Sentence-Bert
SGD	Stochastic Gradient Descent
SQuAD	Stanford Question Answering Dataset
TACOS	Saarbrücken Corpus of Textually Annotated Cooking Scenes dataset
TMN	Temporal Modular Network
TN	Temporal Network
Tree-LSTM	Tree Long Short-Term Memory
VGG16	OxfordNet
VSE++	Visual-Semantic Embeddings

CONTENTS

1	INTRODUCTION	15
1.1	OBJECTIVES	16
1.1.1	General Objective	16
1.1.2	Specific Objectives	17
1.2	STRUCTURE OF THE DISSERTATION	17
2	THEORETICAL BACKGROUND AND RELATED WORKS	18
2.1	THEORETICAL BACKGROUND	18
2.1.1	Neural Networks	18
2.1.2	Deep Learning	19
2.1.3	Convolutional Neural Networks	19
2.1.4	Recurrent Neural Networks	20
2.1.4.1	Long Short-Term Memory	21
2.1.4.2	Gated Recurrent Unit	22
2.1.5	Natural Language Processing	23
2.1.5.1	GloVe	25
2.1.5.2	BERT	26
2.1.6	Modalities	27
2.1.7	Cross-Modality Retrieval	27
2.2	RELATED WORKS	27
2.2.1	Search Method	28
2.2.1.1	Definition of the search sources and parameters for search filtering	28
2.2.1.2	Defining the selection rule for filtering	29
2.2.2	Research Data Extraction	30
2.2.2.1	Video Retrieval	30
2.2.2.2	Moment localization	35
2.2.3	Datasets	40
2.2.3.1	ActivityNet	41
2.2.3.2	Charades-STA	42
2.2.3.3	MSR-VTT	42
2.2.3.4	DiDeMo	43
2.2.3.5	TACOS	43
2.2.3.6	Datasets comparison	43
3	METHODS	45
3.1	THE DATASET	45
3.2	PROPOSED WORKFLOWS	46
3.2.1	The video workflow	47
3.2.1.1	Video frame analysis	48
3.2.1.2	Video frame feature extraction	49
3.2.2	The natural language workflow	51
3.2.2.1	Preprocessing	52
3.2.2.2	Vectorization	52
3.2.3	The embedding space	53
3.2.3.1	Intelligent selection of the negative example video	55

3.2.4	The retrieval workflow	58
4	EXPERIMENTS AND RESULTS	59
4.1	MODEL ARCHITECTURE	60
4.1.1	The Proposed Architecture	63
4.2	INFLUENCE OF EXTRACTED VIDEO FEATURES	64
4.3	INFLUENCE OF GRAMMATICAL CLASSES	65
4.4	THE TRIPLET ARCHITECTURE	66
4.5	CONTROLLED SELECTION OF TRAINING SAMPLES	67
4.6	RETRIEVAL ANALYSIS	69
4.6.1	Ad hoc analysis	73
5	CONCLUSIONS AND FUTURE WORKS	77
	REFERENCES	80

1 INTRODUCTION

The easy access to video recording devices, such as smartphones, handheld cameras, and web cameras, has led to an exponential increase in video production worldwide. A huge percentage of the daily information is recorded in video formats and quickly spread online following a new generation of “always connected” mindset (VORDERER; KLIMMT, 2020). Commercial content, personal videos, official speeches, life logs, and even new types of jobs such as “digital influencers”, which pursue virtual influence by constantly creating new media content using specific channels and strategies to reach the biggest amount of viewers possible (COTTER, 2019), are generating visual information at a very fast pace, getting to thousands of hours of videos per day. On YouTube, a popular online social media platform for video publication, over 720,000 hours of video are uploaded daily (Mohsin, Maryam, 2021).

With that number of new videos being produced, it is becoming crucial to have ways to retrieve meaningful content in an easy and comprehensible way. Popular online search engines (such as Google, Yahoo, and Bing), specialized video search engines (such as YouTube), or even streaming service providers (such as Netflix and Amazon Prime) all have in common that the searches are based on previously indexed texts that are related to the video.

Textual searches that rely on previously produced metadata such as tags, video descriptions, and video names, are an efficient way to retrieve a specific video, but they require huge previous efforts to produce such metadata. This approach, even being extremely effective and widely used, requires that the correct words are used to retrieve the expected video, as the query is based on the match between what is being queried and the available indexed metadata information.

The way a person thinks about a video or scene may work in a wide range of ways, which takes into consideration specific scene memories, phrases spoken during the recorded video, space descriptions, and a mixture of visual, audio, and other modalities that will create the semantic expectation of what is to be queried. When the semantic expectation is clear, but the specific indexed words are not used on the search query, it may produce wrong video retrievals and frustration.

Searches based on object or action descriptions that may be available in a very specific frame range of a large video are even harder to be retrieved if it relies on regular text-to-text searches. Then, given a text that describes what is being searched, how can we retrieve a video

that is not labeled or categorized and so there is no index terms to support the search?

The above-mentioned issues led to a growing area of study aiming at retrieving a video from a given text in Natural Language (NL) using a multimodal approach.

Multimodality is not a new area of study. In linguistics, it has been used for a few decades, described as a way to “make meanings in a variety of different ways” (BEZEMER; JEWITT, 2018). Similarly, in the computation science area, multimodality can be understood as a system that responds to more than one modality or communication channel (JAIMES; SEBE, 2007). Therefore, it is possible to assert that it is expected that the multimodality will make a semantic bridge between several different modalities (i.e., image, audio, text, and videos) reducing the gap between the human perception of the world and the computational relation of different modalities and so, information recovering.

The main issue raised is how to teach a computer system the similarity between two different modalities. One of the most common approaches is to create an embedding space in which two different modalities are compared, teaching the specific models how to identify the similarities among them. Based on this approach, it is possible to minimize the semantic gap, meaning the difference between the previously stated human expectations and the automatic extraction of low-level features between a text query and video content, removing the need to rely on indexed terms to search and retrieve a video (ENSER; SANDOM, 2003). Descriptions of a scene such as “a boy wearing a blue jacket playing outside” will be capable of retrieving videos that best represents what is being queried.

This work addresses the problem of retrieving a ranked list of meaningful videos based on a description of a particular scene using a free text sentence. The existing studies commonly focus on improved ways to extract features from the source video and sentences ignoring the relevance of the learning in the embedding space.

1.1 OBJECTIVES

1.1.1 General Objective

The objective of this work is to propose a method to retrieve a ranked list of meaningful videos, out of an unlabeled video database, based on the video’s semantic content using a NL query as input.

1.1.2 Specific Objectives

- To create a common hyperspace between videos and texts, where modalities with similar semantic meanings are close to each other.
- To perform an in-depth study over a model to observe the influence of parameters and methods in the task of video retrieval based on a NL entry.
- To propose an intelligent method for the selection of the training samples so as to improve similarity learning in the absence of labeled data.

1.2 STRUCTURE OF THE DISSERTATION

This work is structured as follows: Chapter 2 presents the theoretical background needed to perform the experiments in this work as well as the relevant related work found in the video retrieval literature. Chapter 3 presents the methods and data used to perform the video retrieval from NL texts. Chapter 4 presents the experiments and results obtained by this work as well as a comparison and discussion of those results. Chapter 5 presents the conclusions and future works.

2 THEORETICAL BACKGROUND AND RELATED WORKS

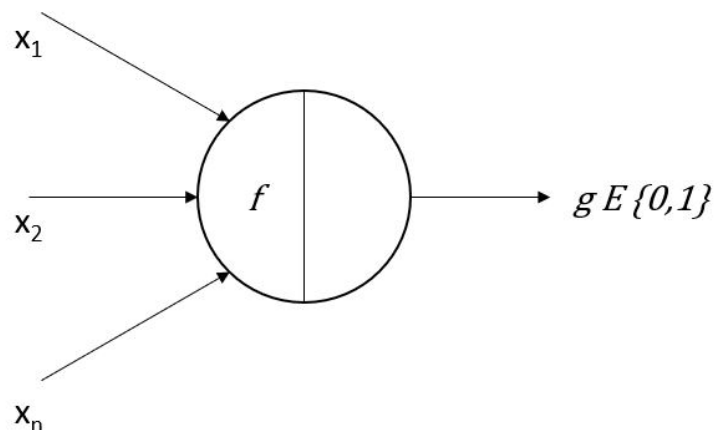
This Chapter first presents the theoretical background needed to perform this work, followed by the most recent literature about the problem, showing the main methods, data and results.

2.1 THEORETICAL BACKGROUND

2.1.1 Neural Networks

NN, also called Artificial Neural Networks (ANN), are computational methods that try to replicate the learning and decision capabilities of biological neural networks. This science field was initially theorized by McCulloch and Pitts (1943) which proposed a mathematical “neuron” model, illustrated in Figure 1, that would act in a binary decision model which accepted binary values as entries. Based on an aggregation function, it would activate or not the neuron in a synchronous sequence. Even thou it seems too simple for today’s computational power and complexity, this model opened the studies for the NN.

Figure 1 – McCulloch-Pitts Neuron representation. The neuron accepts binary values (X_n) that feeds an aggregation function (f) to formulate a binary output (g).



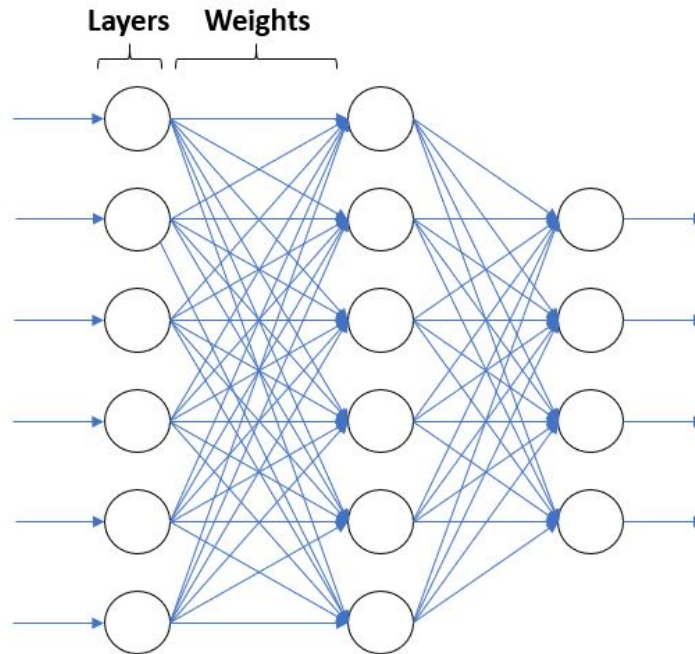
Source: Author.

As the NN evolved, and the computational power increased, newer models were theorized, by connecting several neurons, adding activation functions, layers, asynchronous calculations, and weight sharing, being the base for the Machine Learning (ML) area.

2.1.2 Deep Learning

Using a very complex architecture of many connected NN layers, DL is a sub-field of ML that is capable of abstracting meaningful information from a given input data to automatically learn its representation and make decisions based on adaptable parameters values (weights), which connects different neurons, as shown in Figure 2.

Figure 2 – DL representation. Layers of NN are connected one after the other increasing the number of calculated weights as it gets deeper.



Source: Author.

As more layers are connected, more parameter values are needed to be calculated, requiring a very high computational power to complete its training. To help the training task, it is common to use Graphics Processing Units (GPU) to take advantage of its parallel processing capabilities.

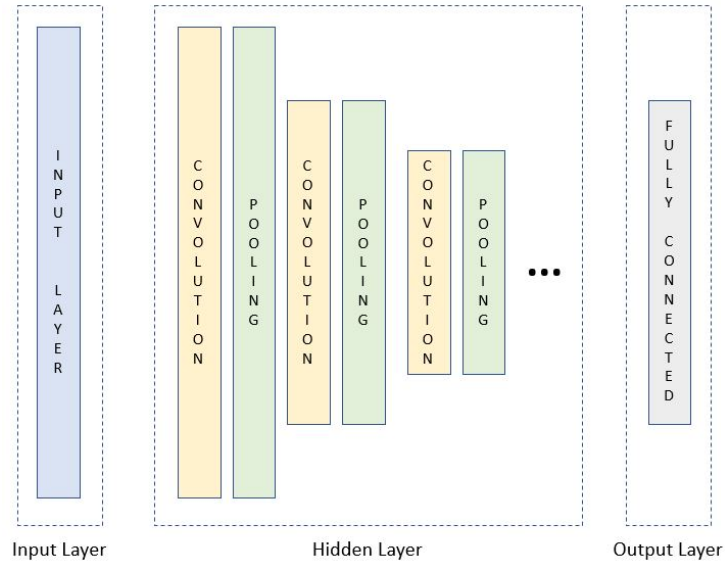
In this work, the two DL architectures used are: CNN and RNN.

2.1.3 Convolutional Neural Networks

Mainly used in Computer Vision (CV) to analyze visual contents, CNN is a DL architecture that can translate a complex grid of information, usually an image, into a feature map representation through convolutional layers. This abstract representation of the original input allows the CNN to learn from the input's most meaningful information to classify the input data

based on the learned parameter values. Figure 3 represents a CNN model.

Figure 3 – CNN representation. The input layer receives the data to be processed. The hidden layer performs convolution and pooling using a kernel to reduce the size of the original data. The output layer performs the classification.



Source: Author.

A CNN is composed of an input layer, one or more hidden layers, and an output layer. The input layer receives the data that feeds the model, usually an image.

The hidden layer may have one or more of the basic layers, convolutional and pooling. The convolutional layer is responsible to perform a mathematical transformation called convolution, using a kernel, or sliding window, which slides across the input object, being activated by an activation function that creates the mapping feature representation. The pooling layer reduces the size of the mapping feature by applying a mathematical pooling strategy, such as an average value or the highest value calculation, also using a kernel. The above process is repeated as many times as the architecture of the model is designed.

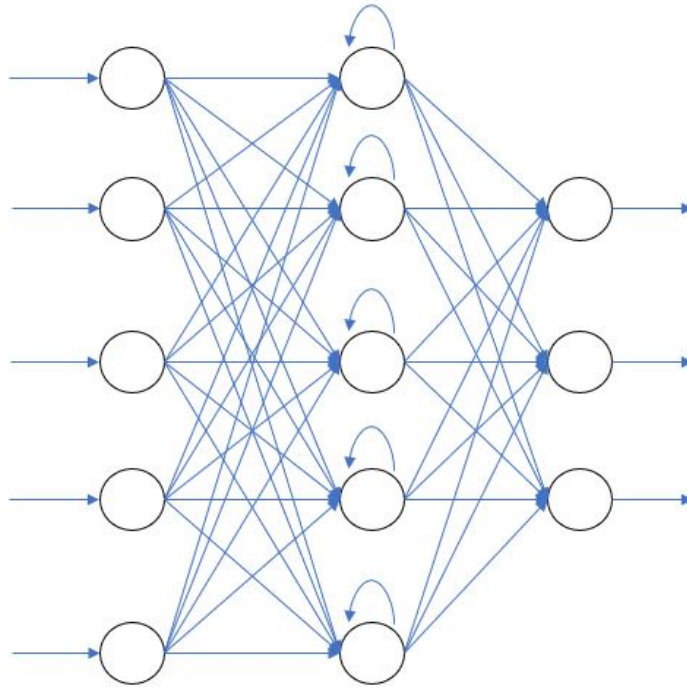
Finally, the output layer uses the created abstract representation to perform the classification using fully connected layers, where each neuron of a layer connects to all neurons of the next layer.

2.1.4 Recurrent Neural Networks

The RNN model was designed to solve the problem where the previous state of the node, or layer, needs to be considered. This allows the RNN models to be capable of handling temporal sequences. A temporal sequence may be a video (an ordered sequence of images) or a

NL sentence (an ordered sequence of words), for instance. Differently from a regular feedforward NN, where the information flows to only one direction between layers, the RNN introduced a connection between nodes that forms a directed or indirect graph, as shown in Figure 4, allowing the information to travel in loops between layers, creating an internal state memory that helps to weights the nodes correctly with its previous temporal information.

Figure 4 – RNN representation. The connection between nodes creates a temporal internal state memory.



Source: Author.

Among the many RNN known models, this work used Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures to manage NL sentences and videos, respectively.

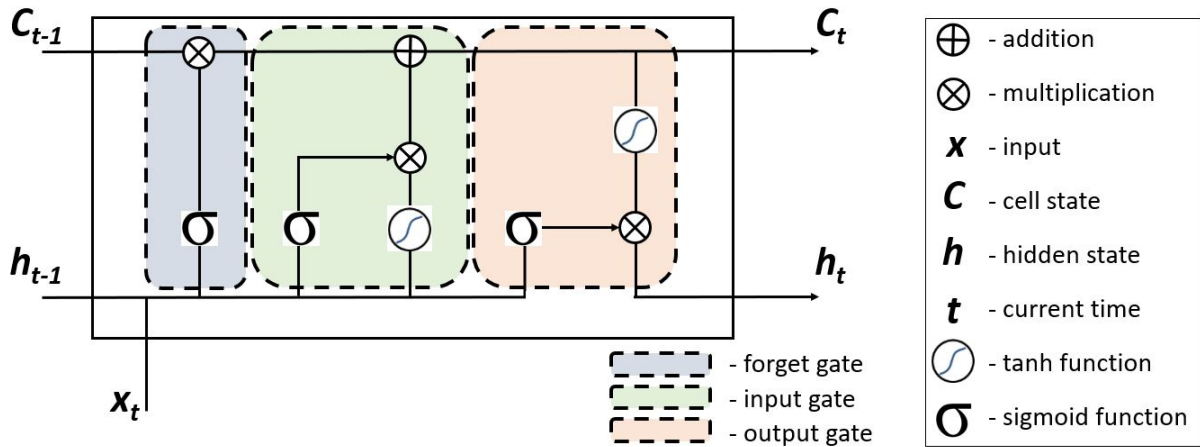
2.1.4.1 Long Short-Term Memory

The LSTM architecture was created by Hochreiter and Schmidhuber (1997) and revolutionized the RNN models with its capability of handling extreme long sequences of inputs, addressing successfully the eventual loss of previous state information with an architecture that is capable of deciding, from a given input x_t at time t , the influence it will have in storing or overwriting the previous state, or memory, information.

Figure 5 shows a representation of the LSTM cell. It receives as entries the input x_t , the previous cell state C_{t-1} and the previous hidden state h_{t-1} . The forget gate is responsible to

decide whether previous information should be kept or thrown away. The input gate, on other hand, decides which new information will be stored in the current cell state updating its internal memory. Finally, the output gate decides which information is going to be sent to the hidden layer.

Figure 5 – LSTM representation of a cell with its internal Forget, Input and Output gates.

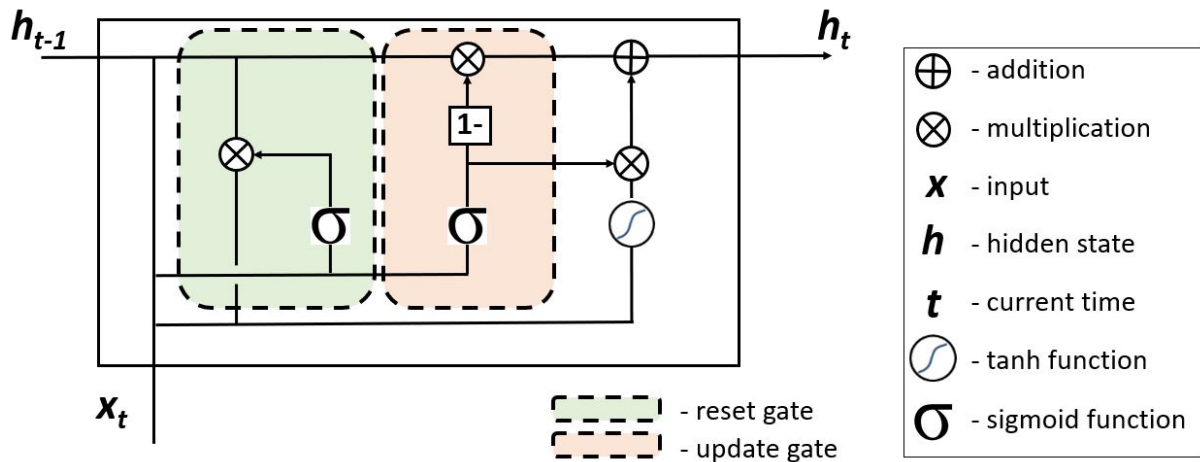


Source: Author.

2.1.4.2 Gated Recurrent Unit

As a variant of the LSTM, the GRU architecture was proposed by Cho *et al.* (2014a) having fewer parameters when compared with the LSTM as it got rid of the cell state, using only the hidden state to transfer information. As shown in Figure 6, its architecture also has only two gates: a reset gate and an update gate.

Figure 6 – GRU representation of a cell with its internal Reset and Update gates.



Source: Author.

The reset gate is used to decide how much of the past information is needed to be forgotten, while the update gate helps the model to determine how much of its past information should be kept and passed along to the hidden state. In the end, both the cell state as well as the hidden state are merged into a single vector.

2.1.5 Natural Language Processing

Human beings use words and sentences to express themselves and to transmit a message or information to a subject by using a specific NL. Natural Language Processing (NLP) is the field of study that merges linguistics and computer science to allow a system to “understand” those messages and information and create intelligence out of it, by extracting meaningful data from speeches and texts, for example.

The text-related NLP activities aim to be capable of performing different activities, such as text classification, summarization, translations, and information extraction. As it is not possible to go straight forward from raw text to an entry that a computational system can accept, such as a NN model, there are necessary pre-process steps that need to be performed before the start of training. The most common steps are:

- **Sentence cleansing:** knowing the fact that a sentence in NL is a wide-spread space and that people now use “emojis” and abbreviations to express emotions, situations, and intentions, it is usually necessary to perform a textual analysis replacing or removing the existing abbreviations, slang and other textual particularities that may interfere in the NLP training.
- **Tokenization:** it is the act of breaking all words and punctuation from a full sentence into smaller units, called tokens, resulting in an array of individual tokens that can be worked and pre-processed individually.
- **Stopwords removal:** stopwords are referred to as the most common words, prepositions, and punctuation that do not influence the meaning of the sentence. Words such as “a” and “the” are frequent in the vast majority of the sentences, with the possibility of happening more than once in the same sentence. Therefore, it can harm the training

of a NN model since it can confuse the model in identifying the most significant words in a sentence. To be capable of removing the stopwords from a sentence, there are publicly available lists of common stopwords from a specific language that can be used.

- **Stemming or lemmatization:** stemming removes the suffix of words, reducing the word into its root, or base, form. Words that share the same root as “likes”, “likely”, “liked” and “liking” are all reduced to their stem “like”. Applying stemming in a text reduces the number of words having the same meaning, thus resulting in more directed training. Similar to stemming, lemmatization also returns the root, or base form, of a word. The main difference between the stem and the lemma is that the latter will always be a valid word, while a stem, depending on the stemming technique used, may not be a valid word. For example, depending on the stemming technique, the stem of “believes” may be “believ”, which is not a valid word, while a lemma will be “believe”.

- **Vectorization:** this step transforms the tokens from their text form to numeric tensors that will be acceptable entries for NN models. The vectorization representation of the sentence can identify each word resulting from the techniques explained before. The problem with the vectorization representation is that it does not give the positional meaning that a complex sentence may require. To do so, each token must have a unique representation that can be achieved with different techniques, as follows:
 - Bag-of-Words (BoW): it is a representation that converts a text into a fixed-length vector, where each position in the vector represents a specific word, and its numeric value represents how many times that word appears in the given text. It is mostly used in classification tasks.
 - One Hot Encoding: Converts a text in a binary array of vectors. The vector has the size of the vocabulary size, where each word is related to a specific vector position having “1” at it while the rest of the vector is “0”. It is useful to represent categorical data.
 - Word Embeddings: It is a floating-point word representation that allows words with similar meanings to have similar representations. There are some famous algorithms to perform this task, such as GloVe and BERT.

2.1.5.1 GloVe

Global Vector (GloVe) (PENNINGTON *et al.*, 2014) is an unsupervised algorithm used to obtain a vectorized representation of words. It has specific groups of available token collections, and the larger one is pre-trained on the Wikipedia Corpus with over six billion tokens.

It uses the Euclidean distance (or cosine similarity) between two words to provide the nearest neighbor words, measuring the semantic similarity of corresponding words, and bringing closer similar words such as lion, tiger, and leopard. Figure 7 shows some unexpected, rare, but relevant words related to the main word “frog”.

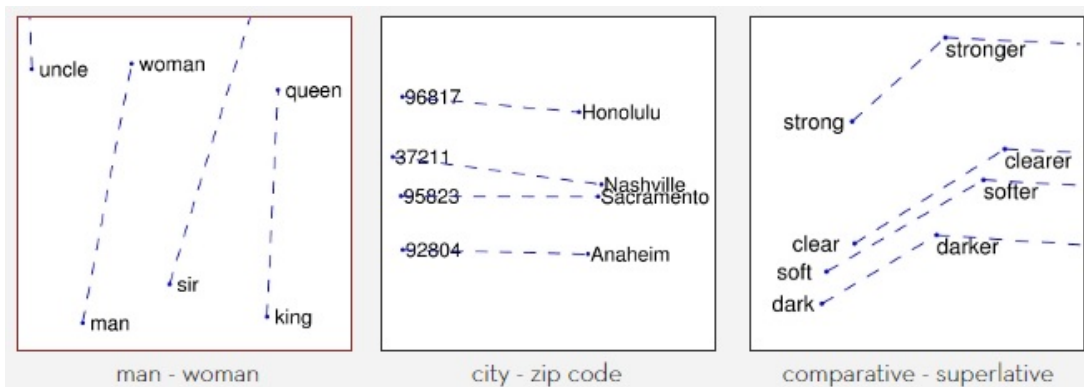
Figure 7 – GloVe nearest neighbor results for the word “frog”.



Source: <https://nlp.stanford.edu/projects/glove/>

GloVe also captures the semantic similarity between groups of words, such as man and woman, boy and girl, and king and queen. It does so by applying equivalent numeric representation in the underlying concept that can group or separate those similar, or distant, words, resulting in linear substructures shown in Figure 8.

Figure 8 – GloVe linear representation of semantic similarity.



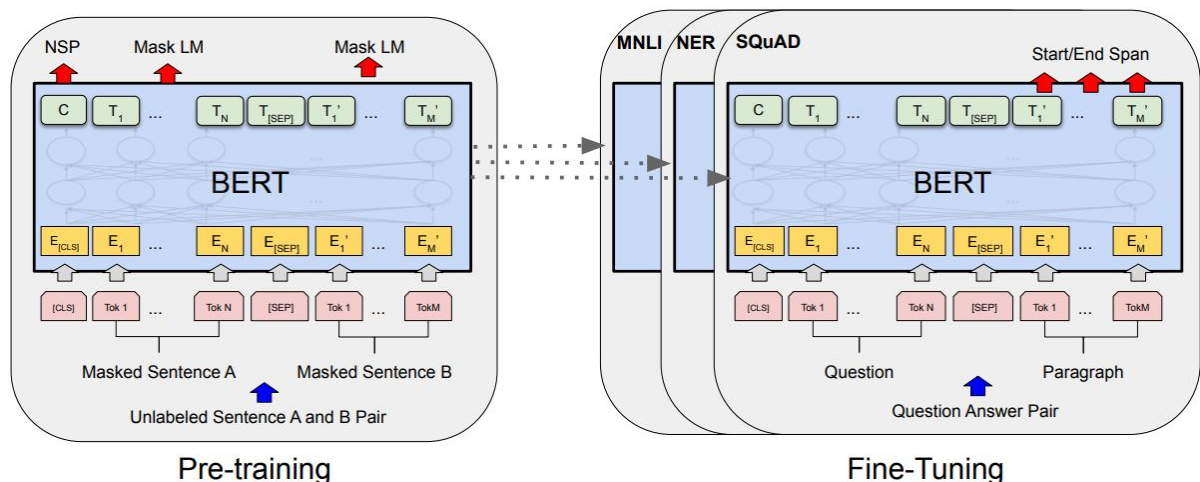
Source: <https://nlp.stanford.edu/projects/glove/>

2.1.5.2 BERT

Devlin *et al.* (2018) proposed the Bidirectional Encoder Representations from Transformers (BERT) as a language representation model that is trained to represent NL sentences using a deep bidirectional model, which can create closer to the human-like understanding of a specific text. In sentences like “a quarter to six” and “ready to go”, the word “to” have different meanings in the full context of the phrase, and that’s what the BERT model tries to address.

Its structure is divided into two major phases: pre-training and fine-tuning, see Figure 9. The pre-training phase was intensively trained using a corpus of over three billion words in free text formats. The fine-tuning phase uses the same structure and parameters from the pre-training phase to fine-tune the model using labeled data.

Figure 9 – BERT phase representations. The pre-trained phase was trained in a large free text corpus, while the fine-tuning phase uses labeled data to fine-tune its learning capabilities, being tested in the major tasks: SQuAD, MNLI and NER



Source: Devlin *et al.* (2018)

BERT makes use of transformers, which are attention mechanisms responsible for learning and understanding contextual relations between a word and its preceding and succeeding words in a text. There are two transformers: an encoder that is responsible for reading the whole NL sentence at once in a non-directional way, allowing it to learn the context of the words in the text, and a decoder, responsible for prediction tasks.

BERT was trained for two main decoder tasks in distinct phases. The Masked LM (MLM) phase hides 15% of the words in each sentence to enrich its word prediction capability. The Next Sentence Prediction (NSP) phase aims to predict if a given sentence is probable based on the first previously known sentence. To do so, it uses 50% of its sentence inputs as identified

pairs, where one is directly related to the next.

2.1.6 Modalities

As mentioned in Chapter 1, modalities can be understood as different ways to make meaning or, in other words, different representations of a semantic subject. For instance, these different modalities could be images, sounds, or text. In this work, the modalities that are used to represent the same subject are:

- **Text:** An ordered sequence of words in a specific language that an individual can understand. By using NLP methods, one translates words and sentences of a NL into a numeric representation.
- **Video:** An ordered sequence of images that creates the temporal information of an action or event through visual information. Computer Vision (CV) methods can process and understand visual data aiming at simulating the way humans do.

2.1.7 Cross-Modality Retrieval

Information retrieval is the action of searching, identifying, and retrieving information that is relevant to a given query.

Cross-modal retrieval refers to the process of retrieving relevant information from a specific modal object that is semantically related to input information from a different modal object, such as retrieving an image related to a text entry.

2.2 RELATED WORKS

The related works presented in this Section are focused on the video retrieval from a NL query. Section 2.2.1 presents the method used to search and filter the selected papers. Next, Section 2.2.2 discusses the related studies and groups the papers in a comprehensible way. In Section 2.2.3 the most used datasets are presented with a brief explanation of each one.

2.2.1 Search Method

Since the subject of interest in this work may present a wide range of potential applications as well as areas of interest, the review of the related work was based on the method proposed by Kitchenham *et al.* (2010). Such a method includes a sequence of filtering steps, as follows:

- a) Define the sources and keywords;
- b) Define a selection rule for filtering;
- c) Define a quality assessment rule for further filtering;
- d) Begin a data extraction procedure;

2.2.1.1 Definition of the search sources and parameters for search filtering

The search for relevant publications was done on well-known scientific sites: IEEE Xplore, Web of Science, Science Direct, and Scopus indexing system. As for the period to be searched, the study focused on the most recent research, limited between January 1st, 2016 to December 31st, 2021. That is, the literature review was limited to the last six years of scientific contributions in the subject of interest. The area of interest, when applied, was focused on the Computer Science field with papers published only in English or Portuguese languages. Finally, the search terms used were: “Video Retrieval”, “Natural Language” and “NLP”.

Considering that those sites allow advanced searches through queries, we defined a string query to be used by the advanced search mechanism on the above-mentioned sources.

Scopus allows a complex search, and the query string used was: *(TITLE-ABS-KEY("Video Retrieval") AND (TITLE-ABS-KEY("Natural Language") OR TITLE-ABS-KEY("NLP"))) AND (LIMIT-TO (PUBYEAR,2021) OR LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019) OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017) OR LIMIT-TO (PUBYEAR,2016)) AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (LANGUAGE, "English") OR LIMIT-TO (LANGUAGE, "Portuguese"))*.

At the Web of Science, the query string used was: *(TS=((Video Retrieval) AND ((Natural Language) OR (NLP))) OR AB=((Video Retrieval) AND ((Natural Language) OR (NLP))) OR AK=((Video Retrieval) AND ((Natural Language) OR (NLP)))) AND LANGUAGE: (English*

OR Portuguese). Indexes=SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH, ESCI Stipulated Time=2016-2021 .

At the IEEE Xplorer, the query string used was: *(("All Metadata": "Video Retrieval" AND ("All Metadata": "Natural Language" OR "All Metadata": "NLP")))* with Filters Applied : 2016 - 2021..

Finally, at the Science Direct site, the query string used was: *Year: 2016-2021 Title, abstract, keywords: (Video Retrieval) AND ((Natural Language) OR (NLP)) Subject Areas: Computer Science.*

A total of 318 papers were returned by the searches.

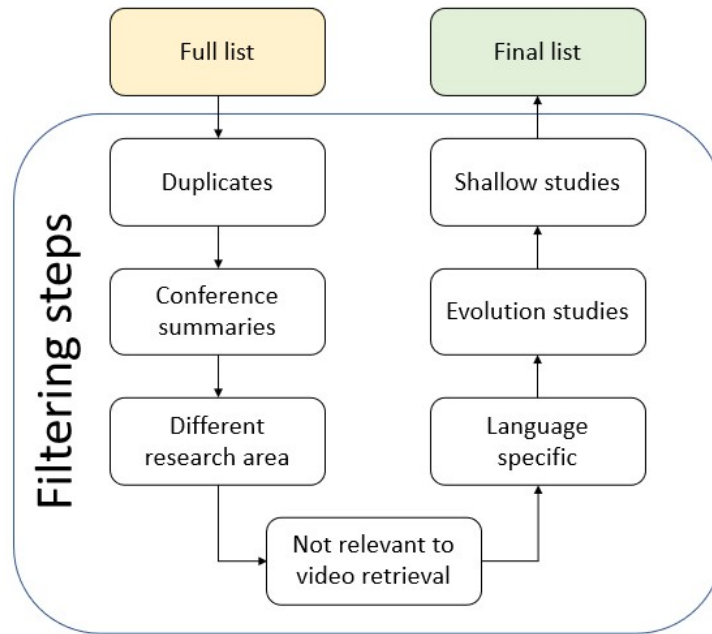
2.2.1.2 Defining the selection rule for filtering

Filtering techniques were applied to the resulting list to clean up and select only the studies and papers that may be of high importance to this research. Figure 10 represents graphically the filtering process followed.

1. The first filter step was to remove the duplicate publications that could be found in the research sources based on the publication title.
2. The second filter step removed conference summaries.
3. In the third filter step, an analysis of the papers' abstract was done to identify the relevance to the topic, removing all papers that were not related to the area of study. The removed topics could be related only to NLP or ontology, audio retrieval, medical related topics, among others.
4. The fourth filter step focused on the relevance to video retrieval only. There are other different areas of study that are also related to video and NLP, such as video description, captioning and tags creation, and question answering. Despite the importance of those areas of study, this work is focused on the NL video retrieval only.
5. The fifth filter step removed the papers that were focused on a specific language solution, such as studies related to NLP in Spanish, Arabic, or Chinese language.
6. The sixth step removed studies that were identified as an evolution of a specific method of the previous paper.

7. In the last step, studies that were too small or too shallow, where the techniques and steps performed could not be identified or reproduced, were removed.

Figure 10 – Visual representation of the filtering steps.



Source: Author.

A total of 41 papers were selected for this study after the filtering process.

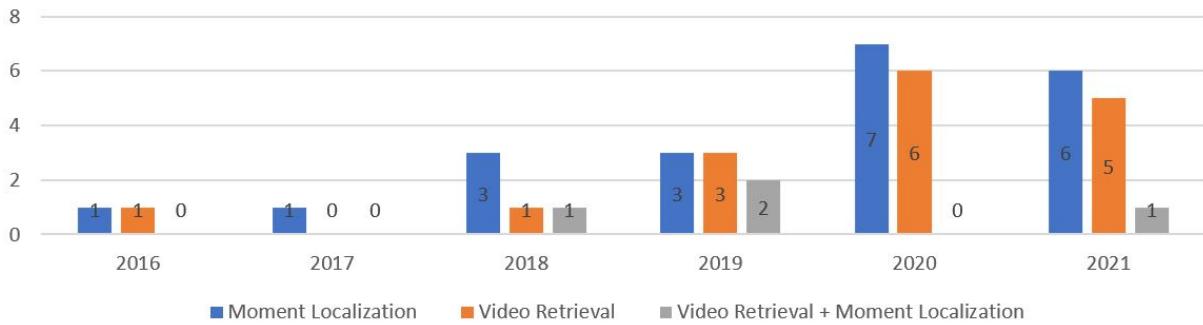
2.2.2 Research Data Extraction

The multimodality between a natural language query and a video is focused on two main areas: (1) the video retrieval, in which the content of the whole video is to be retrieved, and (2) the moment localization/retrieval in an untrimmed video, in which a specific part of the video is to be identified and retrieved. Besides those two main areas, three sub-areas can be focused on, as follows: (a) the improvement of processing of a NL query and its semantics for feature extraction, (b) the improvement of a video processing for actions and objects features extraction, and (c) the improvement of the common embedded space which correlates the natural language query with the video. Figure 11 shows how the subject of study got attention over the years.

2.2.2.1 Video Retrieval

The video retrieval task is, by definition, to return a ranked list of videos that best represents what is being queried from a variety of different videos. To perform this task, the

Figure 11 – A visual representation of how the research areas was focused over the years.



Source: Author.

content of all videos must be taken into consideration, in which the natural language query must retrieve the video, or list of videos, that are most similar among themselves, usually in a ranking format.

a) Video feature-based solutions

Fan and Yang (2020) suggested a solution focused on identifying people and actions. Three vectors of video features were used to strengthen the visual representation to increase the accuracy of video retrieval. The first vector represents people in a cropped tube, the second the scene at a frame level, and the last a combination of the first two. The first two vectors were extracted with a 2D Residual Network - 50 (ResNet50) (HE *et al.*, 2016) and the last one with a 3D ResNet50 network (CARREIRA; ZISSERMAN, 2017).

Similarly, Qi *et al.* (2021) also focused on people tube¹ retrieval, which splits its work in a spatial-temporal encoder-decoder using a Deep Action Proposals (DAP) (ESCORCIA *et al.*, 2016) model to retrieve and locate temporal segments, followed by the extraction of short-term temporal features using a 3D Convolutional Network (C3D) (TANG *et al.*, 2013) and a LSTM network to encode the long-term temporal dynamics. To enrich the visual content, an Attribute Detection Network (AttNet) captures the video-level semantic attributes, based on an OxfordNet (VGG16) (SIMONYAN; ZISSERMAN, 2014) model for feature extraction. That video representation, as well as the embedded text query using a text-CNN (CHEN, 2015), were binarized through a hashing layer, creating binary-coded matrices that can be easily compared.

b) Natural Language feature-based solutions

¹ People tube - A set of connected bounding boxes, or windows, through time-related to the same person.

Focused in enrich the NL query representation to a better learning model, Otani *et al.* (2016) proposed a pre-processing step where the input query was analyzed and fed with static web-collected images to reinforce the visual concepts relevant to the sentence. In this proposal, the top-K web images were downloaded, pre-processed, and merged with the embedded sentence. The resulting feature vector was compared with the video feature vector in an embedding space to minimize the distance between the two models.

The work of Wray *et al.* (2019) presented an approach to enrich the language query by using a Part-of-Speech (PoS) embedded space together with the words embedded space. While in some solutions the query sentence is broken down into the level of the words, this work presented a PoS embedded space which takes into account the combination of verbs, nouns, and adjectives for creating several different and meaningful entities that originate a new space embedding to be combined with the multiple PoS word level embedded spaces, thus finally creating a fine-grained action embedding.

Yang *et al.* (2020) presented an approach in which the sentences are decomposed into tree structures to give complex queries a better semantic meaning. It first extracts features at the word level, using a LSTM network that later feeds a Tree Long Short-Term Memory (Tree-LSTM) (TAI *et al.*, 2015) to create a Latent Semantic Tree (LST) that structurally describes the query based on a memory-augmented node scoring. That structure-aware query representation is compared with an encoded video in an embedding space to find the best matching score based on the cosine similarity.

c) Embedding space-based solutions

With the features extracted from a video and the embedded sentence, Yu *et al.* (2018) proposed a Joint Semantic Tensor (JST) that combines the embedded sentence representation with the features of each frame into a 3D tensor. Later, a Convolutional Hierarchical Decoder (CHD) computes the compatibility score for a pair of multimodal sequences by positively weighting the aligned joint semantics while negatively weighting the misaligned patterns using a ranking loss.

The work from Dong *et al.* (2019) is focused on creating an embedded space that matches two modalities in a concept-free approach. Here, the video is encoded by frame feature extraction that is later globally encoded by a mean pooling that captures the visual patterns,

while the sentences are encoded by a one-hot vector approach. Later, it is fed to a Bidirectional Gated Recurrent Unit (BiGRU) (CHO *et al.*, 2014b) network, creating the global representation of both modalities. The two models are trained together using a Visual-Semantic Embeddings (VSE++) (FAGHRI *et al.*, 2017) for its adaptability to image-text and video-text retrievals.

As an extension of this work, Wang *et al.* (2020b) proposed a solution that uses a Graph Neural Network (GNN) (SCARSELLI *et al.*, 2008) to improve the discriminative ability for finding the positive sample. Here, the fully connected GNN have in its nodes the videos, texts, and query, the relationship between the nodes is made based on the distance between the features. Then, the weight of the relationship between the node and its neighbors' nodes is updated based on the similarities between all neighboring nodes through a message-passing approach for a limited number of epochs getting to a model that is capable of searching both a video from the text as well a text-based from a video entry.

Bansal and Chakraborty (2019) also presented a model that can retrieve a video from a text as well a text from a video. In this work, there is an offline phase that extracts features from both the video and its corresponding captions. Then, both are fed to a two-branch embedding space that learns the similarities between video and sentences based on the Euclidean distance. Finally, an online phase receives either a video or a sentence as input to find its corresponding sentence or video, respectively.

As an improvement on the triplet approach for learning the cross-modality, Akula *et al.* (2021) proposed a model in which the text query is decomposed into verbs and nouns. Then, three examples are compared with the anchor query: a positive, a negative, and a partial. All of them are based on the pair verb-noun, where the positive has the verb and noun (as expected), the negative does not have any of those and the partial is selected by having either the verb or the noun, therefore aiming at to minimize the distance to semantically closer video examples.

To improve the representation of each modality, Dong *et al.* (2021) proposed a multi-level encoding that considers the concept as well as the latent space of each modality to create a hybrid space. The latent space is created by the mean pooling of the features extracted from frames trained with the BiGRU representation of the sentences based on a cosine similarity between the resultant vectors. The concept space is obtained by identifying the most meaningful word concept that is common to all the sentences related to a video and training it with the extracted video representation using the Jaccard similarity approach. The combination of both, the latent and the concept spaces, feeds a dual encoding network that further creates a strong

model for video retrieval.

Taking the advantage of expert models, the works of Liu *et al.* (2019), Sah *et al.* (2020), Gabeur *et al.* (2020) and Chen *et al.* (2021) present solutions in which several different domains of a video (i.e., objects, scene, actions, faces, Optical Character Recognition (OCR), speech, audio) are used to create a full representation of it. Each of those different domains has its features extracted by pre-trained expert models, in which the resultant vectors are then aggregated using different techniques (i.e. pooling). This procedure creates a final representation of the video, composed of several specific features. This resulting representation is then trained in a common space against an embedding representation of the sentence where the similar pairs are closer to each other based on a distance metric.

As an improvement of the expert solution approach, Wang *et al.* (2021) proposed to create a shared center to cluster the local features from multiple modalities related to the same semantic topic. Using this, it is expected that the different modalities help each other to fill the gaps of the same semantic topic, to improve the similarity ranking of the pair text-video.

d) Summary

A summary of the papers analyzed in Section 2.2.2.1 is shown in Table 1. All results listed here used the same metrics, to provide a further comparison with the results obtained in this work. Other works that used different metrics were disregarded. Also, only the results from text-to-video retrieval were selected. Again, experiments that rely on other modalities, such as speech and OCR, were disregarded. The metric being used is the Recall at k ($R@K$) with k assuming the possible values: 1, 5, and 10.

Table 1 – Summary of results of video retrieval works.

Paper	R@1	R@5	R@10	Dataset used
Dong <i>et al.</i> (2019)	7.7	22.0	31.8	MSR-VTT
Wang <i>et al.</i> (2020b)	8.0	23.2	32.6	MSR-VTT
Dong <i>et al.</i> (2021)	11.6	30.3	41.3	MSR-VTT
Yu <i>et al.</i> (2018)	10.2	31.2	43.2	MSR-VTT
Liu <i>et al.</i> (2019)	4.0	14.1	22.4	MSR-VTT
Otani <i>et al.</i> (2016)	7.6	23.4	34.9	YouTube Dataset
Yang <i>et al.</i> (2020)	7.9	20.8	27.8	MSR-VTT
Wray <i>et al.</i> (2019)	14.3	38.1	53.0	MSR-VTT

Source: Author.

2.2.2.2 Moment localization

Focused on a specific moment within an untrimmed video, the video moment localization task searches the whole video for a specific action or moment that happens at a given time. This type of task requires that start and end times of moments or actions are identified in the video to then relate those moments using specific NL queries. These queries, in turn, need to be interpreted so that they can explicitly inform recurrences, actions, etc. Usually, the possible moments identified are ranked based on a similarity distance metric.

a) Video feature-based solutions

To enrich the visual context to identify specific moments of interest within a video, Hendricks *et al.* (2018) proposed a concatenation of three different extracted video features to compose the video representation of a specific moment. The concatenated final video feature is composed of a base moment feature (ground truth), a context moment feature (any other extracted video moment ranked by similarity), and an endpoint feature (extracted video pieces that relate to possible begin and end moments of an action, usually related to words like "first" or "last") that later is related to a NL query in an embedded space. That approach shows how important temporal visual context is for identification.

In addition, Yamaguchi *et al.* (2017) proposed an approach motivated by the visual influence of the moment, with a focus on retrieving a person over time. In the paper, the visual feature is composed of the concatenation of the features extracted from a person's tube and the whole frame. That gives the context of both a specific person and their surrounding space information to the visual embedding. The resulting features are compared with an embedded query in a common space, having the results ranked by similarity.

b) Natural Language feature-based solutions

Understanding the importance of the position of the words in a sentence query to the retrieval process, Barrett *et al.* (2015) presented a work in which the words of the sentence are related to a specific lexicon of 15 words, which allows sharing low-level features and parameters across words, giving meaning to the query and being capable of relating to specific tracked

scenes in the video.

Also focused on decomposing a query sentence, Liu *et al.* (2018c) proposed the decomposition in two components related to a relevant cue. One of them was directly related to the localization of the desired moment, and the other, to the irrelevant cue, not related to that localization. The relevant word is identified per moment of a whole video using a slide window approach. With that, the relationship between the video and the text embeddings is reinforced.

Using a graph approach, Zhang *et al.* (2019d) presented a new model in which a syntactic Graph Convolutional Network (GCN) (KIPF; WELLING, 2016) can represent the words relationship based on its contextual representation. The nodes of the graph are represented by the words themselves, while the edges, first represented by its direct dependencies, are reinforced by the output of a Bidirectional Gated Recurrent Unit (BiGRU) network. That approach resulted in a fine-grained representation of learning capable of exploring the potential relations between a video moment and query contents.

Liu *et al.* (2018a) proposed a Temporal Modular Network (TMN) that uses the Stanford Parser (KLEIN; MANNING, 2003) to obtain the grammatical relationships between words and obtain a parse tree with Part-of-Speech (PoS) tags. With the parse tree, it is possible to obtain new combined nodes, where grammatically equivalent tags are merged, resulting in a reduced tree. The original nodes are then compared individually with the many video moment features in a base model, then creating a map of combined word-level embedding and video encoding. A combination module, corresponding to the combined nodes of mapped child features maps, gives an information map in the compositional hierarchy. Finally, the highest score of the combined module is used to identify the most probable moment of the video.

Also using a tree representation, Zhang *et al.* (2019c) decomposed a sentence using a tree attention network based on a Tree-LSTM. Next, three candidate sentences are created from the original sentence that represents the descriptions of the main event, the context event, and a temporal signal. Those representations are concatenated to be matched in a cross-modal space with a visual focus on visual and location similarities of each proposed moment of the video.

Tang *et al.* (2022) proposed a work that automatically assigns higher weights to query words with richer semantic cues. To do so, the feature representations of the image and the query are analyzed at the frame level. Therefore, the most significant words that represent that particular frame can be found. When all frames are processed, a matrix is created with the frame-words representation.

With a context-aware model that can reduce the noisy background information of a video, Chen and Gu (2021) presented a network that starts with an embedded representation of a sequence of video moments and a given query. Then, it learns the semantic temporal dependencies and applies weights to the most meaningful words per moment, followed by a global interaction module that integrates both videos and learned semantic features. Finally, a foreground re-calibration module identifies the meaningful part of the video related to the meaningful words and cleans up the unnecessary background information, giving a strong moment relation between the modalities.

c) Embedding space-based solutions

To enhance the learning capability of common space, Liu *et al.* (2018b) brought the idea of adding a memory attention layer which, based on a similarity score of each video segment and the query, passes that score to its future moments to memorize the temporal information. Such a procedure leverages the context weights of the important moment features and enhances the moment representations.

Based on a GCN, Zhang *et al.* (2019a) presented the Iterative Graph Adjustment Network (IGAN), capable of encoding complex temporal dependencies that can control the relational information between different moments. In IGAN, each moment is identified as a node in which the information is processed in a cell. In each cell, a residual component from the previous node representation is aggregated with the current node information to produce a representation matrix. This matrix is used to feed the next node, transmitting the temporal information and creating the temporal relationship.

Yu *et al.* (2020) proposed a multi-stream language aggregation model, based on semantic information that can train each moment individually to improve the similarity between a query and a video moment. When all probable moments are trained with the sentence query, the ensemble model combines every single stream to improve its similarity weights.

Using an adversarial approach, Cao *et al.* (2020) proposed a solution in which a generator is set to produce possible video moments, and a discriminator, based on a pairwise ranking model, tries to rank the generated video moments and the ground truth using a triplet approach.

The dual path interaction proposed by Wang *et al.* (2020a) creates both, a frame-to-

candidate representation, and a candidate-to-frame representation. Such an approach takes the advantage of learning boundary information from the whole video representation (frame-level) as well as from the candidate moment-specific information (candidate level). Then, the learned features are cross transferred from one to another. This gives the model awareness of the moment boundaries representation. Consequently, it strengthens the capability of identifying the most probable moment to be retrieved.

Also using two simultaneous networks, Qu *et al.* (2020) takes the encoded video and query features to create two-modal information, where both video-aware sentence and sentence-aware video representations are considered to find the temporal coordinates of the start and end frames of a moment. In this dual network iterative attention module, both the query-video and the video-query share learned weights to better score the probable moment within a video, increasing its accuracy of it.

The model proposed by Jiang and Wu (2021) focused on reducing the divergence of the probability distribution of the video and natural language modalities. It trained both, the specific video, and query subnets to later use transfer-learning techniques to map the extracted features of both, before sending them to a common embedded space. There are three subnets: (1) a video one, (2) a sentence one, and (3) a temporal-based information one. The last one tries to identify the possible available moments. All three models were trained individually and then, merged into a joint model that identifies the distance between the three pre-trained models, and gives a final score to the most probable moment.

Using a one-shot approach, Liu *et al.* (2021) proposed a model that slides over the video and extracts clip features to compare with a text embedding in a common space, where an enhanced cross-modal attention layer is capable of adjusting the weights of the video features according to the text features. Then, a multi-layered perceptron works as a score predictor, and gives a score of the most probable start and end time of the expected video clip.

The model proposed by Sun *et al.* (2021) addresses the problems of having a limited amount of training moments for selection, and insufficient comprehension of structural contexts. Their model is a multi-agent boundary-aware that is focused on finding the best start and end of a probable moment. Here, two agents work together to fine-tune the exact start and end of a moment. One agent focuses on the start point, while the other on the finish. The agents are based on three parameters that are adjusted at each batch run: one for large movement, one for the middle moment, and the last one for small movement.

Focused on a weakly supervised moment retrieval model, Wu *et al.* (2020) proposed a boundary adaptive refinement framework to help the reinforcement learning of temporal boundaries, not using a classical sliding window approach to find the most probable moment within a video. It starts with the extraction of the query features, followed by an extracted clip video feature which is, then, compared with a cross-modal evaluator. As result, a similarity score was computed. Those scores were sent, together with the previously extracted features, to an adaptive layer that keeps the memory of the previous scores by a GRU model followed by two fully connected layers, which works as actor and critic. The critic model provides an estimation value of the current state that the actor model uses to infer the estimation of a gradient. It is used to reinforce the model for the next runs using different extracted clips.

Ma *et al.* (2020) also proposed a model for weakly supervised video moment retrieval. It used a sliding window approach that creates overlapping probable video moment features. Those features, together with a query representation, are fed into a Cascade Cross-Modal Attention (CCA) module that learns the attention weights of probable moments, pruning the irrelevant moments and locating the relevant ones.

Li *et al.* (2021) suggested an approach that creates a 2D representation matrix, by extracting the features of a video with a sliding window and adding the resulting feature vector in a multi-scale 2D Temporal Network (TN). Such a network is later used as input in an embedded space where each proposed moment is jointly merged with the text query representation. Since not all the moments are directly related to the text query, a second step generates pseudo-labels from the top-K scored moment candidates, serving as supervision for training, and enhancing the weakly-supervised model.

d) Video and moment retrieval solutions

The next works perform both, the correct video related to a query, followed by the correct moment retrieval. They are a merge of the two major areas presented.

The Find and Focus model in Shao *et al.* (2018) first performs filtering of the top-K ranked videos, based on a global analysis of their features. Then, the candidate videos are narrowed down to a small representation out of a large number of videos. Then, it applies a clip localization over all the probable video moments, ranking the most meaningful results. The multiplication of the global video retrieval score and the clip localization score presents the most

probable video moment that relates to the query, out of a large number of videos.

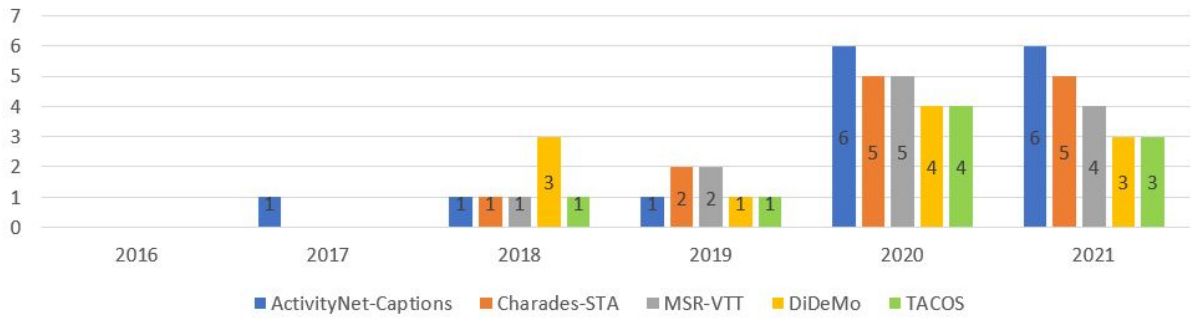
Miech *et al.* (2019) work is focused on the HowTo videos (online available videos that explain visually and with a textual legend, how to perform many different tasks). From that, they perform training on the pair caption-text of a specific moment with its exact video clips moment (as the positive example), and a mixture of the intra-video segment and random inter-video segments (as negative examples), to train a model that is capable of identifying both moments as well as unrelated videos.

The work of Zhang *et al.* (2019b) proposed a feature extraction of key frames of a specific video clip using the Fast Forward Moving Picture Experts Group (FFMPEG) codec (TOMAR, 2006) that is later combined with the global video extracted features, to give context information to each clip. A pair of keys verb-object is identified from that specific video clip and is used to enrich the final visual extracted features. To retrieve the most probable videos out of a list, a list of the key objects is identified from each keyframe and then, they are captioned in words that are used to filter the probable videos. Next, the sentence query is crossed with the enriched visual clip from which features were extracted to retrieve the most probable moment out of videos, ranking the results.

Hou *et al.* (2021) model first identifies the top-K probable videos out of a large list of available untrimmed videos, followed by a moment localization on those selected videos, to retrieve the correct clip out of a list of videos. It merges the features extracted from non-overlapping clips with its text descriptions, such as subtitles or Automated Speech Recognition (ASR) that can vary in time length, depending on the size of the subtitle. With the most probable clip selected, second and third rounds of executions are done to fine-tune the finding of the beginning and end of the clip.

2.2.3 Datasets

Based on the literature review, many different datasets were cited in the published papers. Some of those were developed for specific research, while others are widely used. Based on this fact, we created a rank of the number of times a dataset was used in papers. The top-5 most used datasets cited in Table 2 are shown in Figure 12 as how they are referenced per year. Next in this Section, a brief explanation of these selected datasets will be presented.

Figure 12 – Top-5 most used datasets referenced per year.

Source: Author.

Table 2 – Top-5 most used datasets as identified in Section 2.2.2

Dataset Name	Number of references
ActivityNet	15
Charades-STA	13
MSR-VTT	12
DiDeMo	11
TACOS	9

Source: Author.

2.2.3.1 ActivityNet

The ActivityNet dataset was created by Heilbron *et al.* (2015) and it is a large-scale video benchmark for human activity understanding. It aims at providing a semantic organization of videos focused on human activities. In its first release, the dataset had examples of 7 top-level categories: *Personal Care, Eating and Drinking, Household, Caring and Helping, Working, Socializing and Leisure and Sports and Exercises*. From these top-level classes, a subset of other 203 activity subcategories was used, such as *Painting, Walking the dog* and *Changing Wheel*.

With the activity categories selected, the videos were collected from online repositories, such as YouTube, based on text queries. The selected videos were checked, and those not related to the expected category were deleted. Finally, the videos were labeled using the Amazon Mechanical Turk (AMT) workers. The second round of filtering was done to discard the videos that had labels not directly related to the identified video segment.

The result of this filtering process resulted in a list of 203 activity classes, each of those having an average of 137 untrimmed videos and 1.41 activity instances (or identified moments) per video, in a total of 849 video hours. All videos are shorter than 20 minutes long, with an average length between 5 to 10 minutes. Around 50% of them were recorded in High Definition (HD) resolution (1280×720), and the majority have a frame rate of 30 Frames per Second (FPS).

2.2.3.2 Charades-STA

The Charades-STA (GAO *et al.*, 2017) was proposed to be a dataset focused on temporal activities. It was built on the top of the Combining Human Assessment and Reasoning Aids for Decision-Making in Environmental Emergencies dataset (Charades) (SIGURDSSON *et al.*, 2016), by adding sentence temporal annotations. There are a total of 157 activity categories and around 10,000 videos with multiple video-level descriptions.

The temporal annotations were semi-automatically extracted from original sentences from Charades. The original sentences are long, consisting of sub-sentences connected by a comma, period, and conjunctions, such as *then*, *after* and *while*. Based on this, the sentences were split into sub-sentences by a set of manually collected conjunctions. For each of those sub-sentences identified, the subject of the original sentence was added to the start. Then, keywords were extracted for each activity and matched to the sub-sentences. If they match, a temporal annotation is assigned to the sub-sentence. Finally, a human check is done for each pair of temporal clip annotations and the sub-sentence.

The videos are of an average length of 30.1 seconds long. They have representations of 15 types of indoor scenes, with interactions with 46 object classes. Videos have an average of 6.8 actions per video.

2.2.3.3 MSR-VTT

The MSR-VTT, created by Xu *et al.* (2016), is a large-scale video dataset with 41.2 hours of web-collected videos from all kinds of scenarios. The videos were selected by listing the 257 most popular queries on a commercial video search engine. From there, the top 150 videos were downloaded at their maximum resolution, and the duplicate videos were removed. It has a total of 10,000 video clips and 200,000 clip-sentence pairs, covering 20 categories, and having around 20 natural sentence annotations per clip.

For each video, at most three clips per video were selected (with an average of 2 per video) giving a total of around 30,000 clips. From those, 10,000 were randomly selected and sent to annotation on AMT. Then, all duplicate sentences were filtered, as well as the too-short ones, reaching a total of 20 annotations per clip.

2.2.3.4 DiDeMo

DiDeMo was created by Hendricks *et al.* (2017) and it is a dataset that consists of real-world videos randomly extracted from Flickr videos with a Creative Commons License (CCL). There are a total of over 10,000 personal videos, each one 25-30 seconds long, which were annotated with over 40,000 localized text descriptions.

The videos were segmented into 5-seconds segments to speed up the annotations. Once a moment in the video was annotated, it was validated by another three different annotators. A given moment is added to the dataset as a valid description only if all four annotators agree on it. The major difference presented in the dataset is that, as it has a validation step, the text description describes a specific moment of the video.

As the dataset was built using random real-world videos from Flickr in an open domain with no categories applied, it may consist of open vocabulary, having words such as “*man*” and “*woman*” as well as “*parachute*” and “*violin*”. Also, the dataset allows the descriptions to use the camera movements as a point of comparison having descriptions as “*zooms in on...*” or “*...runs towards the camera*”.

Moments can include any combination of the 5-second long segments, which means that a 30-second long video contains 21 possible moments, which may or may not have text descriptions.

2.2.3.5 TACOS

Focused on cooking activities, Regneri *et al.* (2013) created the TACOS dataset that contains videos of different activities. It has a total of 127 high resolution (1624 × 1224 pixels resolution, at 29.4 FPS) videos of 1-23 minutes long, with an average of 4.5 minutes per video. There are a total of 41 basic cooking tasks, each with 4 to 8 videos.

Each video was annotated with 20 different textual descriptions collected by AMT, leading to 2,540 annotations. A filtering activity was applied to these annotations to remove duplicates or sentences that did not match the correct clip of the video.

2.2.3.6 Datasets comparison

Table 3 shows a comparison between the main features of the top 5 most used datasets.

Table 3 – Top 5 most used datasets comparison

Dataset	Videos	Clips	Sentences	Domain	Source	Hours	Classes
ActivityNet	19,994	$\pm 28,000$	$\pm 28,000$	Open	YouTube	849	203
Charades-STA	9,848	$\pm 67,000$	$\pm 67,000$	Daily Activities	Homes	+83.3	157
MSR-VTT	7,180	10,000	200,000	Open	YouTube	41.2	20
DiDeMo	10,464	26,892	40,543	Open	Flick	+69.4	None
TACOS	123	7,206	18,227	Cooking	Lab Kitchen	+9.2	41

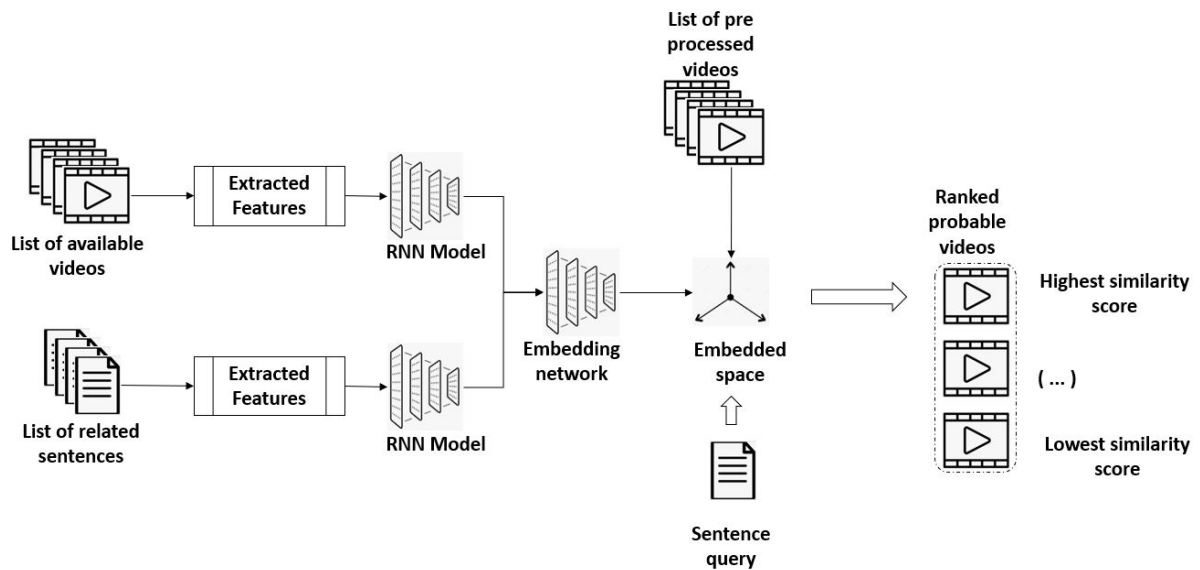
Source: Author.

While ActivityNet have YouTube-sourced HD resolution videos in an open domain space that enhances the training ability, it lacks many description sentences, which was later fixed by the introduction of the ActivityNet-Captions. Similarly, MSR-VTT have YouTube-sourced videos, resulting from the most common searches made in the channel but split into fewer Classes than ActivityNet. DiDeMo introduced the camera as a point of comparison in its descriptions, but it was designed to focus its training on moment localization tasks. Finally, Charades-STA and TACOS are controlled domain types of videos, the first focused on human activities and the second on kitchen actions, aiming its usage to specific training conditions.

3 METHODS

This Chapter presents and explains the methods applied to perform video retrieval using a natural language sentence as a query and the dataset selected for the experiments. Figure 13 shows the architecture of the proposed solution. First, a brief description of the selected dataset is presented. Then, a description of the steps necessary to perform the video retrieval.

Figure 13 – Solution architecture.



Source: Author.

3.1 THE DATASET

The study presented in Chapter 2 allowed us to identify the advantages of each dataset as well as their focus areas. Based on this, the dataset used in this work is that created by Yamaguchi *et al.* (2017), which is a subset of the original ActivityNet dataset, where 5,293 videos of human actions were selected, giving a total of 13.7 hours of videos.

From this list of videos, the authors have selected 6,073 clips to be annotated, some addressing different people from the same video when there was more than one actor in it. Each clip was annotated with 5 related sentences using the AMT approach, where the sentences must have a minimum length of 8 words, be focused on the people's action, and have additional information when possible. A total of 30,365 descriptions were obtained.

In the dataset, there is a variety of actions being described in an open-world domain

with no classes identified. The author split the dataset in Train / Validation / Test following the proportion of 90% / 5% / 5% respectively. Table 4 presents more information about the dataset.

Table 4 – Dataset statistics showing the total amount of available video clips per dataset, the total duration in minutes, the number of people described, and the number of available descriptions.

	Videos	Duration	People	Descriptions
Train	4,734	732 min.	5,437	27,185
Val	276	44 min.	313	1,565
Test	283	46 min.	323	1,615

Source: Author.

The dataset corpus contains a total of 3,785 nouns, 1,982 verbs, 1,451 adjectives, and 262 adverbs. The usage of the words is very imbalanced since there are words that are more commonly found in many sentences, such as those that identify people or colors. Figure 14 presents the top 30 most frequent words of the four classes used in this work.

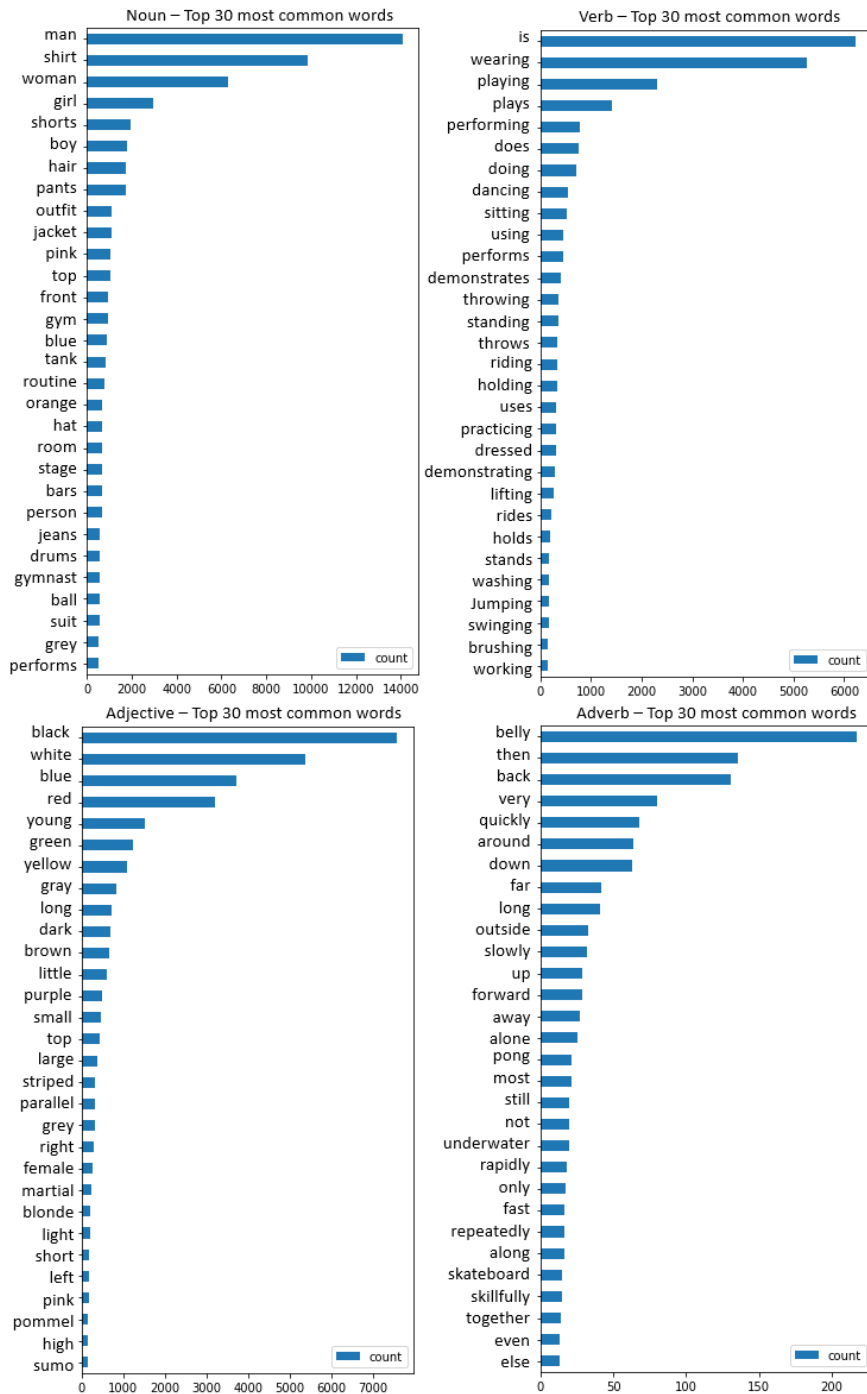
The available videos in the dataset have an average of 121 seconds, from which an average of 13% is annotated. The videos were cut to extract only the annotated clip of the video to have the model learn the relevant information in the training phase, removing all the unnecessary information from it. This procedure resulted in smaller videos to be used in the training and validation phases.

3.2 PROPOSED WORKFLOWS

The solution proposed in this work has three different workflows. The first two are related to the training phase: one is for the training of the video, responsible for extracting the visual and temporal features extracted from a video, while the second is for the training of the NL sentence, responsible for extracting the features of the related sentence. Those two models are merged and trained in a common embedding space in which a function minimizes the distance between the correct pair of video and sentence and maximizes the distance of any other unrelated pair of video and sentence.

The last workflow is for the retrieval, or test, of the videos using the trained models. It is responsible for returning a ranked list of selected videos using the trained models. Here, the features extracted from a NL sentence by the trained model are used to be compared with the features extracted from a list of videos, resulting in the mentioned ranked list, where the smaller the distance, the most probable the video is related to that NL query.

Figure 14 – Top 30 most common words per selected classes in the used dataset Corpus.



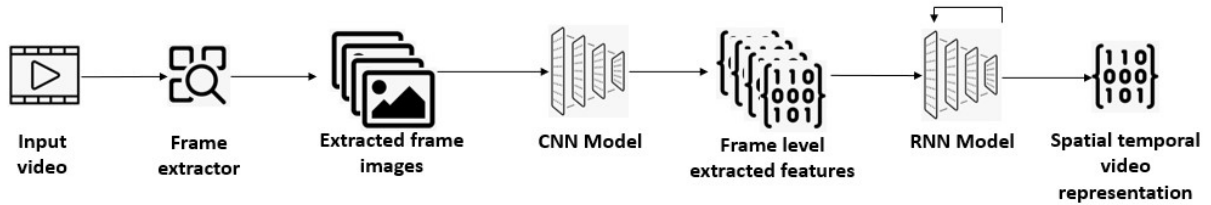
Source: Author.

3.2.1 The video workflow

A video can be understood as a 4D matrix, representing the static image pixels in a 3D matrix of Red, Green and Blue (RGB) information, and another 1D space representing the temporal information.

A two-step approach was used to collect all that information. First, using a CNN model to extract frame-level features, followed by a RNN model that gives the temporal information. Figure 15 shows the end-to-end approach from a selected video to a spatial-temporal feature representation.

Figure 15 – Video feature extractor workflow. Frame-level features are extracted using CNN models which, in turn, are used as input to a RNN model, thus resulting in a spatial-temporal representation of the video.

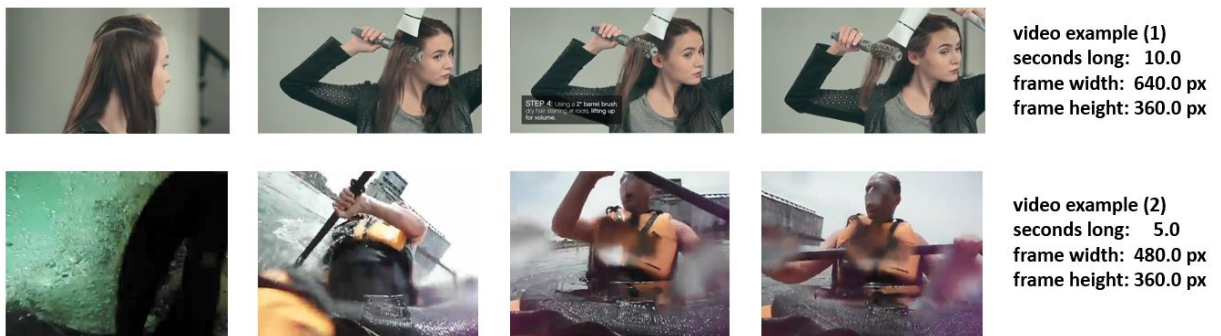


Source: Author.

3.2.1.1 Video frame analysis

A clipped video clip may have different sizes in shape (width and height), as well as different time lengths, as they are in the original videos. Figure 16 shows the comparison between two video clips with their related information. Therefore, some analyses were done on every resulting video clip to understand its features and normalize it, before feeding the CNN model.

Figure 16 – Video clip comparison to exemplify the difference in lengths and frame shapes.



Source: Author.

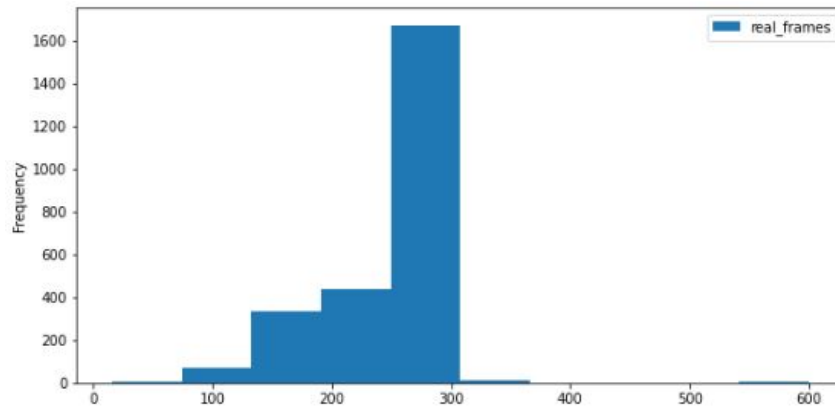
Understanding that the duration of a video (V_d) can be achieved with a simple calculation of the number of available frames (fr_{num}) multiplied by the FPS rate (fps) (Equation 1), the first analysis done was to identify the average number of frames in the video clips to choose the ideal number of frames per video to be used in the training that causes the minimal information loss.

$$V_d = fr_{num} * fps \quad (1)$$

Given a video file, a frame extractor was used to collect the image representation of each frame. To do so, the well-known Open Source Computer Vision Library (OpenCV) (BRADSKI, 2000) library was used for image manipulation.

In Figure 17 it is easy to realize that most of the video clips have a maximum number of frames of around 300. Consequently, that was the number of frames used in this work to represent a video clip and to feed the model for training. Any other video clip with more than 300 frames was limited to this upper bound. As for the videos that have less than 300 frames, an approach of masking was performed.

Figure 17 – Number of maximum frames per video clip.



Source: Author.

Masking a video clip is a way to use a support array to inform which of the extracted frames should be considered in the training and which should not. This can be achieved by creating fixed-length vectors filled with zero values that are updated with the frame-extracted information. For each updated vector position, a support vector indicates if the vector position is valid or not. In the end, a pair of feature and mapping mask vectors are created. Algorithm 1 illustrates the process, and Figure 18 shows an example of a pair of feature and mask vectors expected for a given video.

3.2.1.2 Video frame feature extraction

The frame level feature extractor can be done using many known CNN models, such as VGG16 and ResNet50. As the dataset used in this work does not have identification of classes and does not have a large example representation for training from scratch, a transfer learning

Algorithm 1 – Video feature extraction with masking.

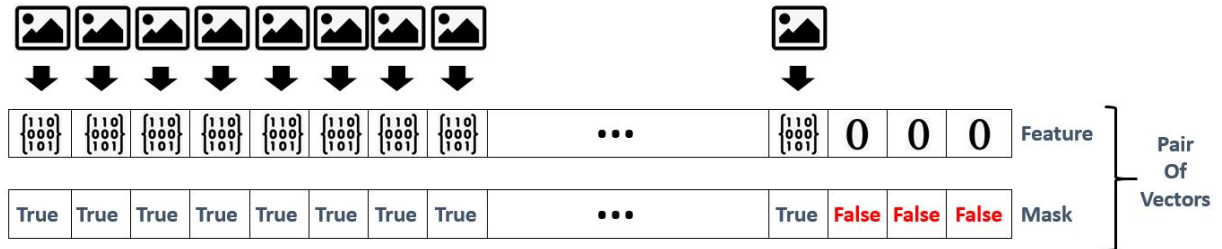
```

1:  $MaxFrames = 300$ 
2:  $Vector_{feature} = \text{New Vector}[MaxFrames]$  of Zeros
3:  $Vector_{mask} = \text{New Vector}[MaxFrames]$  of False
4: for  $iteration$  in  $V_d$  extracted frames do
5:    $Vector_{feature}[iteration] = \text{CNN frame extraction}$ 
6:    $Vector_{mask}[iteration] = \text{True}$ 
7:   if  $iteration > MaxFrames$  then
8:     Stop
9:   end if
10: end for
11: return  $Vector_{feature}$  and  $Vector_{mask}$ 

```

Source: Author.

Figure 18 – Pair of feature and mask vectors. Each frame of a video has its extracted features filling a position of the vector. Parallel, a mask vector informs that the vector position is a valid one.



Source: Author.

approach (GUTOSKI *et al.*, 2021) was used. Four models were compared at the frame-level feature extractors, all pre-trained on the ImageNet dataset: VGG16 (SIMONYAN; ZISSERMAN, 2014), MobileNet-V2 Network (MobileNetV2) (SANDLER *et al.*, 2018), ResNet50 (HE *et al.*, 2016) and Inception-V3 Network (InceptionV3) (SZEGEDY *et al.*, 2016).

The architecture of all four models can be compared in Table 5. VGG16 is a very well-established model, and it is largely used in the studies presented in Section 2.2.1. However, it has the caveat of being extremely complex with several thousands of parameters, besides being the oldest of the feature extractor models tested. ResNet50 is the second largest model, with fewer parameters than the previous, but with a large number of layers. InceptionV3 is as large as ResNet50. Finally, MobileNetV2 is the newest and the light-weight model of them all, with the lowest number of trainable parameters.

Table 5 – CNN models architecture comparison.

Model	Parameters	Layers	Output vector	Reference
VGG16	138.3 M	16	512	Simonyan and Zisserman (2014)
ResNet50	25.6 M	50	2048	He <i>et al.</i> (2016)
InceptionV3	23.8 M	48	2048	Szegedy <i>et al.</i> (2016)
MobileNetV2	3.4 M	53	1280	Sandler <i>et al.</i> (2018)

Source: Author.

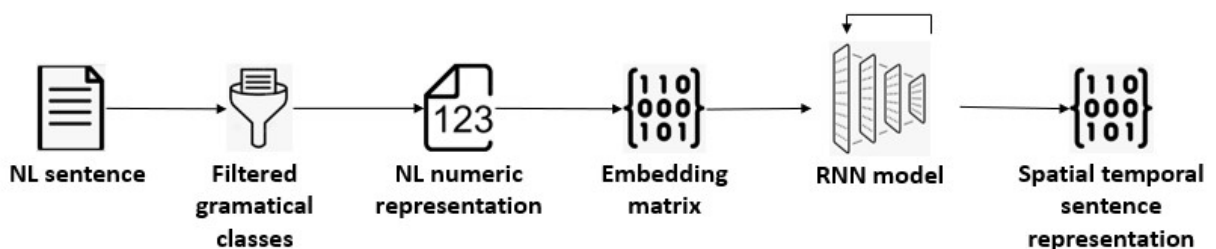
The usual process to extract the features, regardless of the model, is to execute the CNN extractor without its top layers (the dense layers that perform the classification). Since this is not a classification problem, but an extraction of features, this top layer is not needed. Also, since there are limitations in input shapes for the extractor models used, the largest common shape was used to standardize the size of the frames. In this work, all frames were resized to the shape of 224 pixels for both the width and height of the image using OpenCV.

With the frame features extracted and the map masking to inform which frames should be considered in training, a RNN model was run using the frame-level features as input to give the temporal perception of the video. A GRU model was used as the RNN model for being a smaller model, as shown in Section 2.1.4.2, expecting it to use less memory and be faster than other larger RNN models.

3.2.2 The natural language workflow

The description of the scenes, the subjects, and the actions happening in a video is presented as NL sentences. Therefore, it is a key point to be capable of mapping and extracting the correct features from the NL sentences. The first step is to select only the words that have relevant meaning to the sentence. The filtered words must be converted to a numeric representation that is used to define an embedding matrix which, in turn, contains all known words from the corpus of the dataset. Finally, a RNN model gives a positional-temporal meaning to the whole sentence. Figure 19 represents the workflow for NL feature extraction.

Figure 19 – NL feature extraction workflow. Meaningful words from the sentence are converted to numeric representations that are used as input to a RNN model, resulting in a positional-temporal representation of the sentence.



Source: Author.

3.2.2.1 Preprocessing

As explained in Section 2.1.5, there is a logical sequence of steps that must be taken to prepare and convert the NL sentence into a numeric tensor that can be understood by a NN model.

The Natural Language Toolkit (NLTK) (LOPER; BIRD, 2002) library was used for the preprocessing steps required for this task. This library provides a powerful NLP suite using an Application Programming Interface (API) that helps in most of the steps, also having over 50 corpora and lexical resources in many different languages.

From the previously presented steps in this work, the sequence used for the NLP is: “sentence cleansing”, “tokenization” and “stopword removal”. Neither stemming nor lemmatization was performed to not cause the words to be unexpectedly misinterpreted by their reduction. The recognition of similar words was transferred to the embedding phase.

Given the fact that a complex sentence is constructed by different words from different grammatical classes, or Part-of-Speech (PoS), first, it is necessary to be capable of identifying the words that have a semantic meaning for the correct understanding of the sentence. In this work, the PoS of interest are those belonging to classes of *Nouns*, *Adjectives*, *Verbs* and *Adverbs*. It is understood that with the words from these grammatical classes it is possible to capture the relevant meanings of a specific sentence from the subject, i.e., its description, the action being taken, and any subtle specific description of the action. NLTK is also capable of identifying the correct grammatical class of a word based on its full sentence, as shown in Table 6.

Table 6 – NLTK grammatical classes conversion used.

Grammatical Class	NLTK Classes
Noun	NN, NNS, NNP, NNPS
Adjective	JJ, JJR, JJS
Verb	VB, VBD, VBG, VBN, VBP, VBZ
Adverb	RB, RBR, RBS

Source: Author.

3.2.2.2 Vectorization

With the sentences tokenized, the conversion of the tokens into numeric representations followed a simple mapping approach in which each unique token from the whole identified corpus receives an exclusive numeric representation following an increasing sequence.

Next, since the tokenized sentences have different sizes with different numbers of tokens, a normalization of the lengths of the sentences had to be done. To do so, all sentences were padded based on the biggest tokenized sentence, filling the difference in sizes with “0” (zeros). That process is called NLP masking, and, like the video approach, has the purpose to indicate to the NN model which vector position to use or not.

For the embedding of the words and the creation of the embedding matrix, the GloVe algorithm explained in Section 2.1.5.1 was used. As a result, it was able to generate an embedding matrix that can represent each word’s meaning, semantic relationship, and the context in a dense vector representation format.

As the sequence and the relationship of words matter for this work, a BoW approach may not be suitable, generating a lack of dependencies between words. In a BoW approach, semantic relations would be lost, causing the sentences *"boy rides a bicycle"* and *"bicycle rides boy"* to have the same words and so the same numeric representation. To avoid this situation, a RNN model was used to create the positional and temporal meaning of a sentence. The LSTM (HOCHREITER; SCHMIDHUBER, 1997) model was used for this purpose.

3.2.3 The embedding space

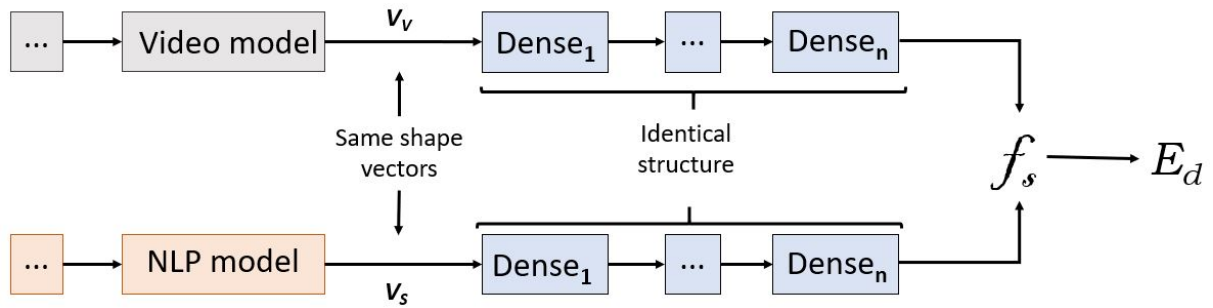
The results of the video and NL feature extractors are a pair of vectors that need to have the similarity metric calculated between them. To do so, the solution used was to develop a Siamese, or Twin, neural network.

The Siamese neural networks are composed of two, or more, identical NN as inputs that share weights and parameters to have the models to be trained in a joint embedded space. They have to learn how to maximize the relevant features to reduce the distance between similar examples.

Since the modalities under comparison are of different natures, that is, NL sentences and videos, the models have different base models, but identical dense layers. Figure 20 presents how the Siamese models can be created from different modalities.

In this work, the Siamese network used to train the models was the Triplet-Loss function (SCHROFF *et al.*, 2015). This model uses the concept of an anchor point in which two input samples, a positive and a negative, are compared. The objective is to minimize the distance between the anchor and the positive sample at the same time, maximizing the distance between the anchor and the negative sample, as is illustrated in Figure 21. In this work, the anchor is the

Figure 20 – Siamese model showing the video and NL different input models sharing the same dense layers to produce similar results that can be compared in f_s Siamese function to calculate the Euclidean distance E_d .



Source: Author.

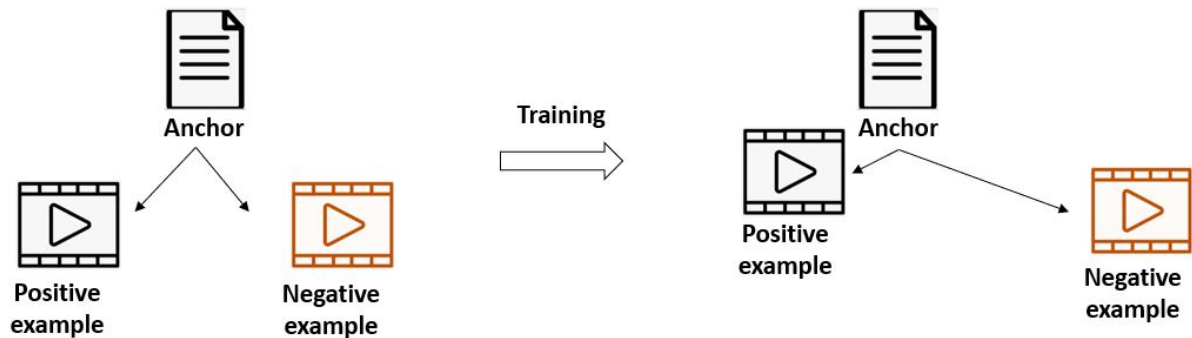
NL query, the positive example is a video mostly related to the NL query, and a negative example can be any other unrelated video, as shown in Figure 22.

The similarity distance between the query and videos is calculated in the embedding space, using Equation 2.

$$T_l = \max(0.0, E_d(A_s, V_{ip}) - E_d(A_s, V_{if}) + a) \quad (2)$$

where T_l is the calculated Triplet-loss, E_d is the Euclidean distance, A_s is the feature vector extracted from the anchor sentence, V_{ip} and V_{if} are the feature vectors extracted from the positive and negative videos, respectively, and a is an alpha value that is responsible for fine-tuning the calculation of the loss.

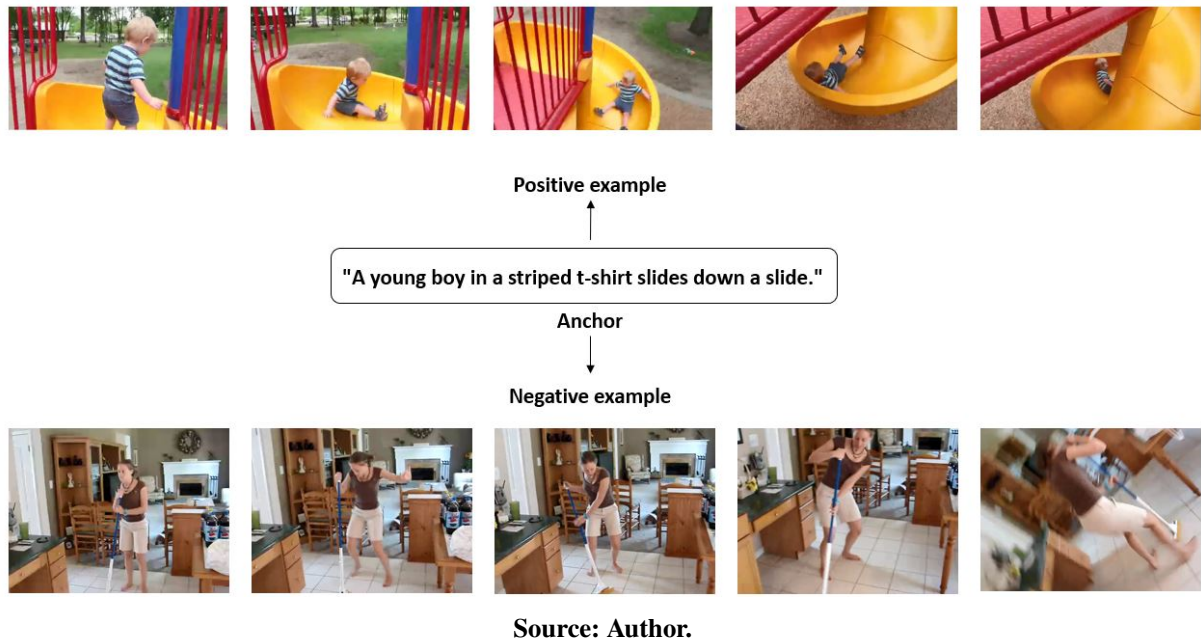
Figure 21 – Triplet-loss function illustration. Based on an anchor, a positive and a negative example, the model is trained to minimize the distance between positive examples and maximize it, otherwise.



Source: Author.

Given the fact that the dataset used in this work is not categorized, there is, still, a challenge to identify the optimal negative samples for a given anchor. That is because of the semantic meaning of a video. For example, given an anchor as (A) “A man working in an office.”

Figure 22 – A NL sentence as an anchor, and a pair of videos as a positive and a negative example.



and two videos, which descriptions could be (Ex1) “*A man playing soccer.*” and (Ex2) “*A woman working in an office.*”, which one would be the best negative sample?

To tackle this problem, two different techniques were proposed in this work: (1) random selection of an example, fully transferring the responsibility of finding the subtle differences to the neural network model, and (2) calculating the distance between the anchor sentence and the videos-related sentences (known in Train and Validation datasets) to find the sentence with the largest distance to the anchor and use its the related video as a negative example, in a way to improve the negative example selection for training the models.

3.2.3.1 Intelligent selection of the negative example video

As explained in the previous Section, the selection of the negative example may cause a direct impact on the training result, since its features are directly related to the resulting value of the loss function.

A random selection of those negative examples may or may not bring a video that is not similar to the anchor sentence. Therefore, an intelligent process to choose the video based on its descriptions was built to reduce randomness in such a procedure by finding a more suitable negative example. This process follows a specific sequence of steps taking advantage of the known video descriptions available at the train and validation datasets.

(1) Sentence selection

As shown in Section 3.1, the dataset used has five different sentences for describing each video. This number of videos is intended to provide a diversity of video descriptions to enrich the training, however, it can also mislead the selection of a negative example. The proposed solution is to create a sample of sentences and select one sentence per video, allowing a direct comparison of descriptions.

(2) Embedded procedure of the sample sentences

Once the sentences were sampled, the embedding of all the NL descriptions was done, so to allow further mathematical comparison between texts.

As presented in Section 2.1.5.2, the BERT model can translate the full meaning of a text, taking into account words and their surrounding words to give a more meaningful understanding of the video description. The Sentence-Bert (sBert) framework (REIMERS; GUREVYCH, 2019) was used to facilitate the use of BERT model. sBert helped with the encoding of the sample data and the target (or anchor) example texts. By the end of this processing step, a vector containing the extracted features of all sample data is available for comparison.

(3) Ranking process

From a given anchor text, related to a positive example video, this step created a ranked list of the less similar texts, related to the anchor, from the sample data. To do so, the anchor text was encoded using sBert to have a vector representation of the same shape as the embedded sample data. Then, a one-to-one comparison was done, from the anchor data to all the available embedded texts in the sample. This was accomplished by using a cosine distance between the vectors to find their similarities.

Those calculated distances were sorted from the lowest to the highest values and, then, the data samples were ranked and sorted by the text with the lowest similarities. Figure 23 shows some examples generated from given anchor examples.

(4) Video selection for negative example

Figure 23 – Ranked negative example texts. The ranked sentences in the middle column are ranked as opposite to the anchor sentence in the left column. The related distance similarity found is listed in the right column.

Anchor Text	Ranked negative example texts	Similarity
A man in a white hat and brown shirt chugs an alcoholic drink in a glass bottle.	A woman performing in a field event for a competition	-0.25503045
	A woman is running on a track field.	-0.23138478
	A woman plays with a small dog on a porch.	-0.21961395
	A woman in a green shirt doing yard work	-0.21412009
	A woman wearing black is mowing the lawn.	-0.19409771
A girl braids her shoulder length hair back inside home	A man in a green shirt plays the bongos outside.	-0.19225362
	A shirtless man with a hat doing flips outside	-0.18804446
	A matador wearing blue and gold is taunting a bull.	-0.17862234
	A man wearing a yellow shirt performs a flip on a slack line outside.	-0.17707598
	A man in a black shirt and a green hat is break dancing outdoors.	-0.17095244

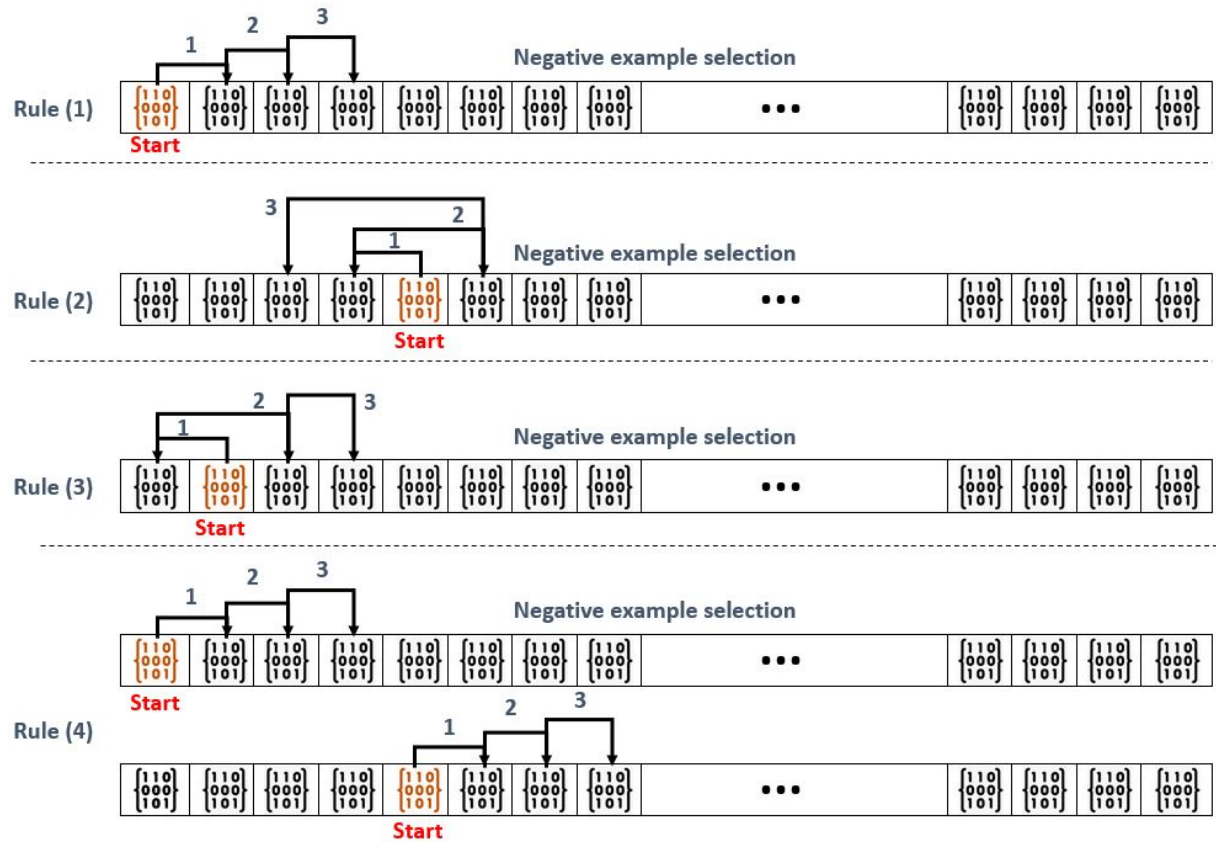
Source: Author.

Once the ranked negative sample data was created, as well as the anchor and positive and negative video examples, the last step was the generation of the dataset used for training and validation. For this task, the ranked sample data were used to identify their related video.

From there, each anchor text can be related to one or more negative video examples based on the ranked sample data, starting from a controlled desired position of more or less similar videos. Keeping in mind that a video has five related descriptions, the same negative example video must not be repeated. Therefore, the iteration over the ranked sample data must be sequential while at the same positive video example.

Also, considering that one may want to provide more than a single negative example for the same anchor positive example, the datasets followed a logical sequence of selection and movement over the ordered list of negative examples. This sequence considers these rules: (1) if starting from the first or last position, it must progress by selecting the next available video in an ordered sequence; (2) if starting from any point in the middle of the list, it must stay as close as possible to the starting point, interleaving examples from the top and bottom of this starting point; (3) if it starts from a middle point and reaches any of the edges of the list, it must change its behavior and progress as an ordered sequence, where available examples are available; (4) at the change of a positive example of the same anchor, it must continue from the last selected negative example, not repeating any negative video. An illustrative example of the rules can be seen at Figure 24.

Figure 24 – Example of the rules applied to select the negative examples. The steps taken to select four negative examples for each positive video are shown, from the starting video followed by three selection loops.



Source: Author.

3.2.4 The retrieval workflow

With both the video and NLP models trained in the embedded space, the retrieval workflow uses the learned common characteristics between the two modalities to rank videos, out of a list of unseen videos, that better matches the NL sentence.

In the video portion of the workflow, the pre-trained video model is used to predict the resulting vector of each video in the list. This will generate a list of embeddings to be compared.

In the NLP part of the workflow, a specific sentence is preprocessed following the steps shown in Section 3.2.2.1. Next, it uses the tokenizer map and the GloVe embedding matrix, explained in Section 3.2.2.2, to convert the tokens into a padded numeric vector. Then, the pre-trained NLP model predicts the sentence and generates the NL embedding to be compared.

Finally, for each embedded video, the cosine distance is calculated with the embedded NL sentence. That distance is the metric used to rank the similarity between the NL sentence and the video, such that the smaller distances mean high similarity between both modalities.

4 EXPERIMENTS AND RESULTS

This Chapter explains the experiments done, their objectives, procedures, and parameters, as well as the results obtained. First, Section 4.1 presents the experiments done to compare the influence of deeper layers and most common optimizers that defined the architecture model used for further experiments (See Chapter 3). Section 4.2 compares the influence of the model used to extract video features (see Section 3.2.1). Next, Section 4.3 compares the influence of the grammatical classes used to create the sentence embeddings (see Section 3.2.2). Section 4.4 presents the experiments related to the parameters of the triplet-loss embedded space (see Section 3.2.3). In Section 4.5, a sequence of tests was done to observe the impact of using a controlled selection of negative example videos (see Section 3.2.3.1). Finally, Section 4.6 presents a qualitative analysis of the retrieval results achieved in the previous experiments.

All experiments were run on a server with an Intel Core i7 processor (8 cores) at 3.30GHz and 32GB RAM, and two TITAN Xp GPUs, running the Ubuntu 18.04 operating system. For implementing the DL models and all experiments, the following software were used: Python 3.8, Tensorflow (ABADI *et al.*, 2015) 2.7, and Keras (CHOLLET *et al.*, 2015) 2.7.

The models were trained for 30 epochs using the dataset explained in Section 3.1. An early stopping approach was tried, but that made the model stop its training in less than five epochs, so it was decided to remove it and force the 30 epochs of training. Since there is no categorical classification for the data in the dataset used, all results were measured using R@K of the top-k in the retrieval results. Recall is a metric that calculates the fraction of relevant instances that were retrieved from the top-k videos returned from the retrieval given a query.

During the execution of the initial experiments, we noticed that each training took an average of 4.5 hours to complete. An automated testing procedure was developed to be capable of performing the necessary experiments in an optimized way. Such automation creates both models, trains and validates them, and executes a retrieval over the test dataset to calculate the metrics and collect random results examples. All the results are reported in a text file for later analysis. The automation works by accepting as input a configuration file that informs all the necessary configuration parameters needed per test, including:

- Test identification;
- Encoder model being used;

- Video extraction model;
- Video RNN model;
- NLP grammatical classes;
- NLP RNN model;
- Number of dense layers for the encoder, video, and NLP;
- Dropout for video and NLP;
- Encoder optimizer and its configurable parameters;
- Encoder alpha value for fine-tuning.

The same configuration file also receives a list of the test identification that are expected to be executed, allowing control of either linear or parallel experiments.

4.1 MODEL ARCHITECTURE

Chapter 3 presented the overall architectural model that was designed to meet the objectives of this work. Some empirical tests were done to adjust the model to its best performance. The main objective of the experiments reported in this Section is to obtain an efficient structural model to be used in further experiments.

To have a fair comparison between the architectural proposals, all video features were extracted using MobileNetV2, and then fed a GRU RNN, while all NLP process was done using all four identified grammatical classes listed in Table 6, with GloVe embedding that fed a LSTM RNN model. At the triplet-loss embedding space, the negative example needed was randomly chosen.

The main questions these experiments aim to answer are:

1. What is the impact of deeper dense layers?
2. Does stacked RNN layers improve the temporal and positional perception of the model?
3. Would a faster optimizer perform better in a cross-modal task?
4. What is the impact of the dropout factor to avoid overfitting?

Starting from a model composed by a single GRU RNN layer for video, a single LSTM RNN layer with GloVe embeddings for NLP, both followed by a single dense layer, this ground architecture was referenced as the minimum base needed to execute the proposed work. The Stochastic Gradient Descent (SGD) optimizer was used with a learning rate of 0.01.

To answer Questions 1 and 2, two executions were performed over the ground model by first adding two extra dense layers that kept the same size of nodes across the layers. Then, another stack of RNN layers was included in both, the video and the NLP models. Table 7 presents the results of this experiment for $k = \{1, 3, 5, 10\}$.

Table 7 – Top-k results in percentage for model architecture comparison.

k	Ground Model R@K	Extra Dense R@K	Stacked RNN R@K
1	0.79	10.67	0.39
3	2.37	24.90	1.18
5	3.95	33.99	1.97
10	7.90	49.80	3.95

Source: Author.

It was expected that the ground model would not perform well, but the results obtained in the stacked RNN model were much lower than expected. That may be due to the level of abstraction obtained in stacked RNNs, which led to a lack of semantic information in different modalities. Because of that, the next experiment used the model with two extra dense layers.

The next experiment compared the impact of reducing the number of neurons by half in each extra dense layer, to take advantage of the convolutions in a NN model. Table 8 shows the results of the experiment.

Table 8 – Top-k results in percentage, for training with reduced layer's size.

k	Same layers size R@K	Reducing layers size R@K
1	10.67	3.95
3	24.90	9.48
5	33.99	15.81
10	49.80	24.90

Source: Author.

Even though the convolutions should keep the semantic meaning of what is being trained, the results of the approach with a reduction in the number of nodes per layer were worse than keeping the size across layers. The explanation might be related to the fact that in this cross-modality problem, the bigger resultant vector might carry more information that can be used for comparison, not losing, or transforming, any feature that could be relevant to the comparison.

Based on these results, the answer to Questions 1 and 2 is to use an architecture that has two dense layers, but only a single RNN layer for video and NL. Also, the size of nodes should be kept the same across the dense layers. The model with the best performance was used in the next experiments.

With the architectural model defined and focused on answering Question 3, different optimizers were also tested to compare the convergence of the model during training and their impact on the results.

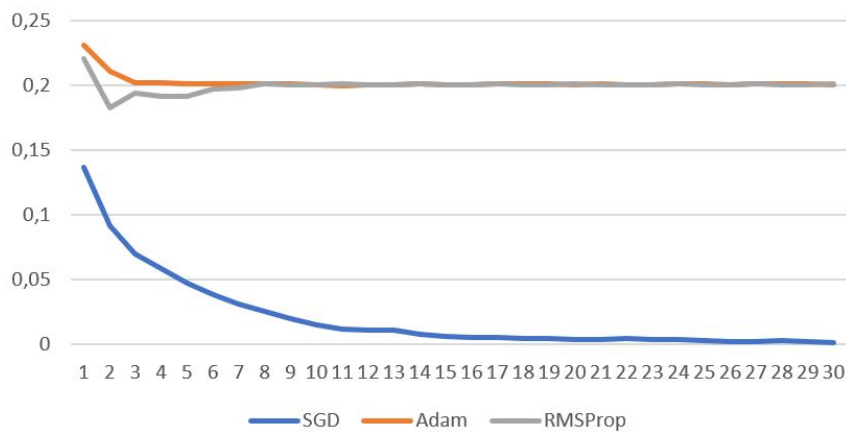
Other than the traditional SGD, the optimizers Adaptive Moment Estimation (Adam) and Root Mean Square Propagation (RMSProp) were tested. Both Adam and RMSProp are optimizers that should show a faster convergence, improving the learning speed. Table 9 shows the results for this experiment, while Figure 25 shows the related loss curve.

Table 9 – Top-k results in percentage for comparing the optimizers.

k	SGD R@K	Adam R@K	RMSProp R@K
1	10.67	0.39	0.39
3	24.90	1.18	1.18
5	33.99	1.97	1.97
10	49.80	3.95	3.95

Source: Author.

Figure 25 – Loss curve for comparing the optimizers.



Source: Author.

Observing the results and the loss curve it is clear that the Adam and RMSProp led to a premature convergence during training, in a quite small number of epochs. This possibly indicates that the model is overfitting.

Also, comparing the Adam and RMSProp columns in Table 9 with the stacked RNN column in Table 7, a pattern can be identified, indicating that those small numbers may be related

to a random selection of results in the Validation dataset, showing that the models did not learn how to correlate the different modalities.

Given the fact that it is a cross-modal model with no categorical classification, a faster optimizer may quickly lead to an overfitting scenario, thus answering Question 3.

Still related to overfitting, the next experiment tried different dropout values for both, video and NLP models. The dropout variation (d) increased from small to large, demonstrating that the variation applied can affect overfitting. Table 10 shows the experiments with various d values in the video workflow. Following that, and keeping the best value found for the video workflow, Table 11 shows the results for the variation of d , but in the NLP workflow.

Table 10 – Top-k results in percentage for video dropout (d) variation.

k	$d = 0.1$ R@K	$d = 0.3$ R@K	$d = 0.5$ R@K	$d = 0.7$ R@K
1	10.67	9.48	9.48	0.39
3	24.90	20.94	18.57	1.18
5	33.99	31.22	26.87	1.97
10	49.80	45.45	40.71	3.95

Source: Author.

Table 11 – Top-k results in percentage for NLP dropout (d) variation.

k	$d = 0.1$ R@K	$d = 0.3$ R@K	$d = 0.5$ R@K	$d = 0.7$ R@K
1	10.67	9.09	9.88	7.90
3	24.90	21.73	22.92	17.78
5	33.99	38.06	29.64	26.48
10	49.80	42.68	48.61	39.13

Source: Author.

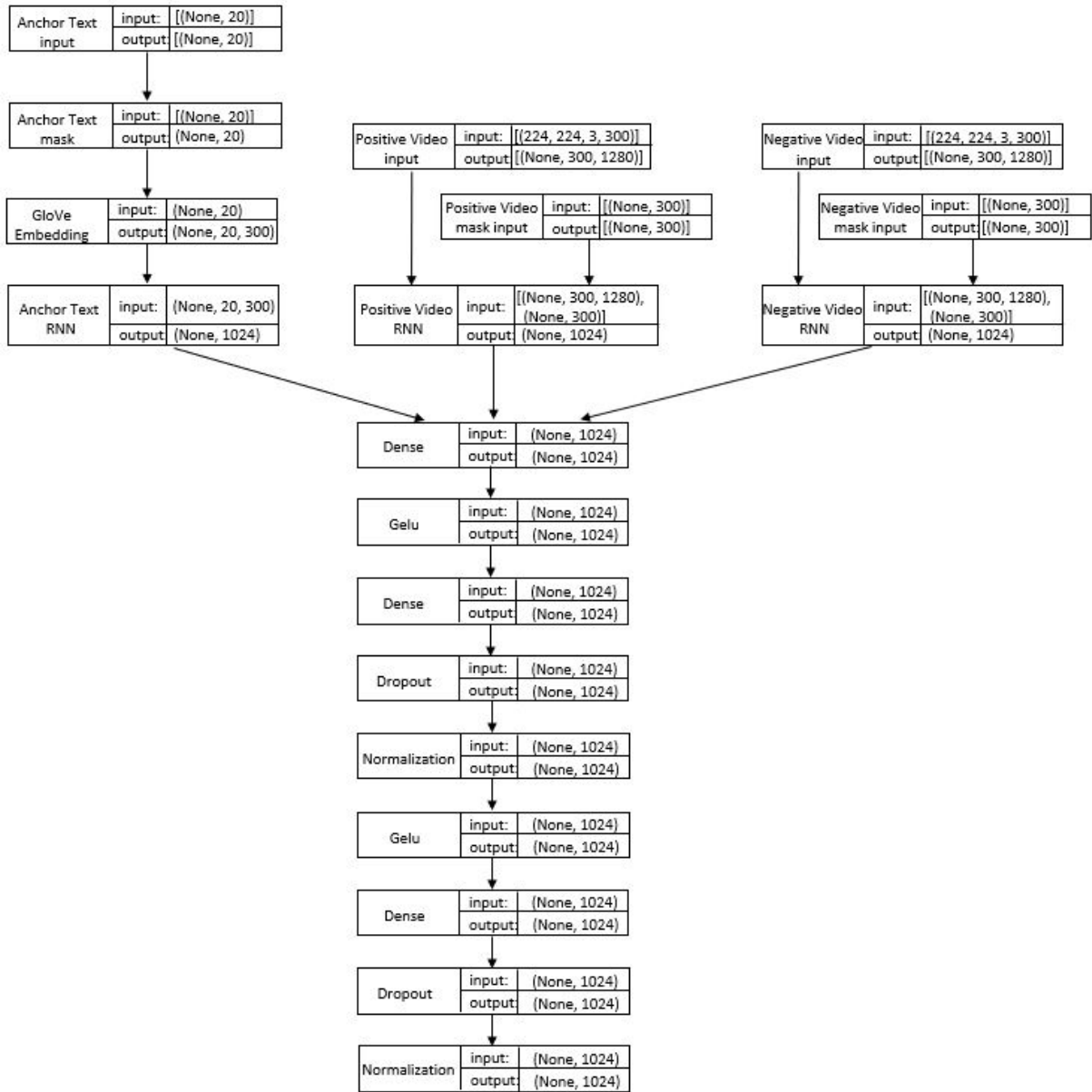
A negative impact on the results was found as the value of d increased. Most probably, this is related to the fact that the higher the value of d , the more random input values were altered to “0”, losing meaningful information for the cross-modality comparison, similar to what happened in the layer size reduction experiment. So, Question 4 is answered.

4.1.1 The Proposed Architecture

Based on the results of the previous experiments, we propose the Triplet network architecture shown in Figure 26. It accepts three different feature vectors as input, one for the NL anchor, and two for positive and negative video examples, followed by a sequence of three same-sized dense layers. The optimizer defined was SGD, with a learning rate of 0.01 and

momentum of 0.5, and the dropout was set to 0.1 to avoid overfitting and not cause an impact with the loss of information.

Figure 26 – The triplet function architecture.



Source: Author.

4.2 INFLUENCE OF EXTRACTED VIDEO FEATURES

In this Section, the experiments were focused on verifying the influence of the video feature extraction on the model.

As explained in Section 3.2.1, the video workflow is composed of a frame-level extraction phase that is responsible for collecting the content information of an image, followed by a

RNN model that gives the temporal perception of the video as a sequence of extracted frame information.

In Section 4.1 a single layer RNN was shown capable of acquiring the necessary temporal information for the cross-modality training. Therefore, the frame-level phase is the object of testing in this Section. The question to be answered with this experiment is:

5. Given the fact that there are differences in the number of parameters and depth in the CNN extraction models, how do they impact the training results?

As presented in Table 5 (Section 3.2.1.2), there are significant differences between the four tested models, such that VGG16 the largest and MobileNetV2 the smallest. There is also a difference in the size of the output vector, which, based on the results of the experiments performed in Section 4.1, the amount of information for cross-modality comparison seems to be relevant. All four models were tested using the architecture previously defined using a transfer learning method previously trained on the ImageNet dataset. The outcomes can be compared in Table 12.

Table 12 – Top-k results in percentage for video feature extractor models.

k	MobileNetV2 R@K	VGG16 R@K	InceptionV3 R@K	ResNet50 R@K
1	10.67	5.13	3.95	6.71
3	24.90	12.25	11.85	18.97
5	33.99	20.55	16.99	28.45
10	49.80	37.94	28.06	43.08

Source: Author.

The best results were obtained with the MobileNetV2 model, followed by ResNet50. Curiously, InceptionV3 presented the worst results.

MobileNetV2 was a model recently introduced, which may indicate that the method used to extract features can be more effective than the older approaches. It does not have the largest output vector, but the extraction method may collect the most meaningful and relevant information from the image from a cross-modal perspective, answering Question 5.

4.3 INFLUENCE OF GRAMMATICAL CLASSES

The NL is the entry of the query, the base of the cross-modal retrieval task. Section 3.2.2 introduced the model that translates a given sentence to a numeric vector based on words, or tokens, from selected grammatical classes listed in Table 6.

Since NL is very complex, having a variety of nuances and semantic meanings, the question to be answered in this experiment is:

6. What is the influence of verbs and adverbs in the content description of a video?

Understanding that a query is composed of several descriptions of a specific scene and that the description may rely only on nouns and adjectives (i.e., “*a boy with a blue jacket*”), this experiment ran tests using only the grammatical classes of nouns and adjectives, as well as a completed test with nouns, adjectives, verbs, and adverbs. The results are shown in Table 13.

Table 13 – Top-k results in percentage for training using different grammatical classes.

k	Nouns, Adjectives, Verbs and Adverbs	Nouns and Adjectives
	R@K	R@K
1	10.67	7.90
3	24.90	20.94
5	33.99	30.83
10	49.80	42.29

Source: Author.

Observing the results, it is clear that describing actions using verbs and adverbs has a positive impact on the extracted features. It may not be related to the number of words used, given that there is no discernible difference in the maximum size of usable tokens identified when only nouns and adjectives are used (the maximum size of usable tokens is 17 tokens long) and when both verbs and adverbs are used (maximum size of usable tokens is 20 tokens long), but it is most likely related to the fact that a verb can help to identify the action occurring in the video.

Then, the answer to Question 6 is that verbs and adverbs represent a meaningful part of the information in the description of a NL sentence, needing to be taken into account in further experiments and analyses.

4.4 THE TRIPLET ARCHITECTURE

As explained in Section 3.2.3, the triplet-loss function used in this work relies on an anchor (the NL query), a positive example (the NL query related video) and a negative example (any other video not related to the NL).

The embedding space is trained based on the distance from the anchor and the two example videos, fine-tuned by an α parameter that can increase or decrease the training rate, as shown in Equation 2. The tests performed in this Section aim to answer the following question:

7. What is the direct impact on the retrieval results based on the value of α ?

To answer the question, a sequence of tests was accomplished by varying the value of parameter α in the range 0.05, 0.2, 0.5, 0.8, 1.0. The results are available in Table 14.

Table 14 – Top-k results in percentage for Triplet-loss impact analysis over α variation.

k	$\alpha = 0.05$ R@K	$\alpha = 0.2$ R@K	$\alpha = 0.5$ R@K	$\alpha = 0.8$ R@K	$\alpha = 1.0$ R@K
1	9.88	10.67	3.95	0.39	0.39
3	22.92	24.90	13.43	1.18	1.18
5	34.38	33.99	21.34	1.97	1.97
10	47.43	49.80	33.20	3.95	3.95

Source: Author.

On the one hand, it was discovered that as α increases, the model's learning capability is negatively affected, such that no learning occurs for values of α equal to or greater than 0.8. On the other hand, when α is too small, it has minimal relevance to the learning, probably related to the fact that the triplet function relies on distances, and a small value of α may have a small influence on the distance between the anchor and its examples, leaving the similarity calculation more subjective, thus answering Question 7.

4.5 CONTROLLED SELECTION OF TRAINING SAMPLES

All tests performed up to this point used a random selection of the videos for negative examples related to the anchor text. This means that the video can be of any subject scene other than the positive-related ones. As introduced in Section 3.2.3.1, the following tests demonstrate the impact of having a controlled selection of negative examples, using the cosine distance between the BERT extracted values of the original sentence and all the other video-related sentences available in the training dataset.

Considering that there will be an ordered ranking list of sentences and related videos compared to the anchor being processed, it is easy to manipulate the selection of the desired distance from the anchor using the positions of this ranked list. Here, the tests performed aim at answering the following questions:

8. What is the impact of controlling the negative examples for a given anchor?
9. Considering its impact, what is the most significant distance from an anchor?

10. Considering its distances, is it better to use a single distance group or a mixture of different distances for a given anchor?

The first set of tests was focused on analyzing the impact of the distance of the negative example from a given anchor. It was expected to see how the variation of the distance affects the results obtained at the tests, starting from the most distant negative example, and gradually moving the distance up to reach the closest negative example available.

To do so, the positions of the ordered list of negative examples were used to identify a given distance to be tested. The ordered list's total length was divided into five folds, resulting in starting positions that may represent distances (d) of 0%, 25%, 50%, 75%, and close to 100% similarity to the anchor video. It can be understood that position 0 of the ordered list is the most distant from the anchor possible, while the last position is the closest distance possible from the anchor.

As observed in Table 15, the examples that are too far ($d = 0\%$) or too close ($d = 100\%$) to the anchor presented results that had no impact on the training of the model, leading to random results. That can be explained by considering that examples too far away are so different that it does not require any effort for the model to identify their differences, while examples too close are very similar, making it hard for the model to identify differences between the original positive video and the negative example, resulting in a poorly trained model in both cases.

The best-observed similarity distance between the negative example and the anchor relies on the range of 25% to 50% distance, probably because in this range the negative example videos have differences but yet some similarity, even if it may be small, causing the model to be able to learn how to differentiate both videos.

Table 15 – Top-k results in percentage for the impact of controlling the distance d of negative examples for a given anchor.

k	$d = 0\%$ R@K	$d = 25\%$ R@K	$d = 50\%$ R@K	$d = 75\%$ R@K	$d = 100\%$ R@K
1	0.39	1.58	1.18	0.79	0.39
3	1.18	3.55	3.95	1.97	1.18
5	1.97	5.53	5.92	3.55	1.97
10	3.95	8.30	8.69	7.11	3.95

Source: Author.

The last round of tests aimed to observe how a balanced distribution of examples could affect the results. These tests used a distribution of examples in two and three groups of distances.

When using two groups of distances, the weights were applied at the edges of the list. Having four negative examples to each positive example, the balanced weights tested used a

proportion of one distance to the three closest example videos, and then three distances to one close example video.

When using the three groups of distances, the weights were applied at the edges and in the middle of the list. The first test had one for each of the three distances, having a proportion of three negative examples to each positive video, while the second test had two at each edge and six in the middle, having a proportion of ten negative examples to each positive video.

Table 16 – For negative examples for a given anchor, top-k yields a percentage of the balanced distribution of controlled distance d . The percentage distributions indicate the weights applied considering the left number as the most distant to the anchor and the right one as the closest.

k	25% - 75%	75% - 25%	33% - 33% - 33%	20% - 60% - 20%
	R@K	R@K	R@K	R@K
1	0.39	0.39	0.39	0.79
3	1.18	1.18	1.18	1.97
5	2.37	1.97	1.58	4.34
10	4.74	4.74	5.92	8.69

Source: Author.

The results observed at Table 16 confirmed that a distribution focused on the middle examples has better results than the ones focused on the edges of the ordered list. Also, when comparing with Table 15, it is observed that the introduction of edge examples to a test focused on the middle (50% distance) confuses the model, reducing the R@K values. This can be explained by the fact that the introduction of edge examples is the same as forcing “noise” to the datasets in a relevant proportion of 40% of their examples, causing more harm than good to the model

The controlled choice of the negative examples used in the triplet architecture has a direct impact on the results of the training of the model. A wrong choice of negative examples can cause the model to behave differently than expected, making it a delicate process to fine-tune the training. Therefore, Question 8 is answered. Also, the tests confirmed that the best-observed similarity distance between the negative example and the anchor relies on the range of 25% to 50% distance, with a caveat that adding edge examples to the training dataset may cause “noise” which results in a worse model, answering both Question 9 and Question 10.

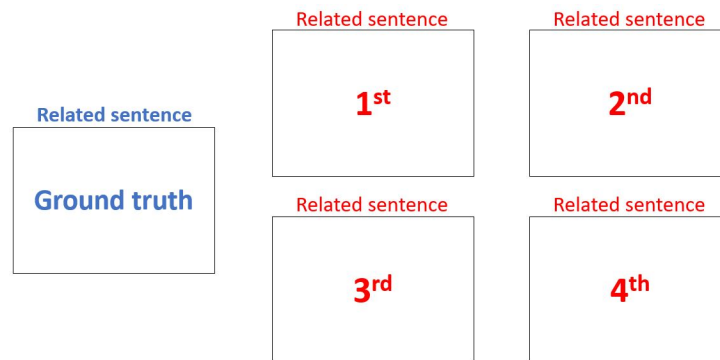
4.6 RETRIEVAL ANALYSIS

Other than the numeric analysis, the cross-modality field requires strong semantic analysis to identify the most meaningful video subject that the model learned, related to a NL sentence. In this Section, visual and semantic analysis was done over the results obtained in the retrieval workflow, focusing on answering the following questions:

11. How did the model learn to identify the similarity? Is it based on the scene or the action?
12. Given a specific sentence with verbs, nouns, and adjectives, will the model focus its results on nouns, adjectives, or verbs?
13. How does the model behave when it is confronted with a word for which it was not trained?

To better illustrate the results, a grid of images was created for comparing the ground truth image and the 1st, 2nd, 3rd and 4th ranked results. The Figure 27 explains how to read it.

Figure 27 – Grid results explanation – the left square is the ground truth, while the squares number 1, 2, 3 and 4 represents the 1st, 2nd, 3rd and 4th ranked results. At the top of each square, the related NL sentence is shown.



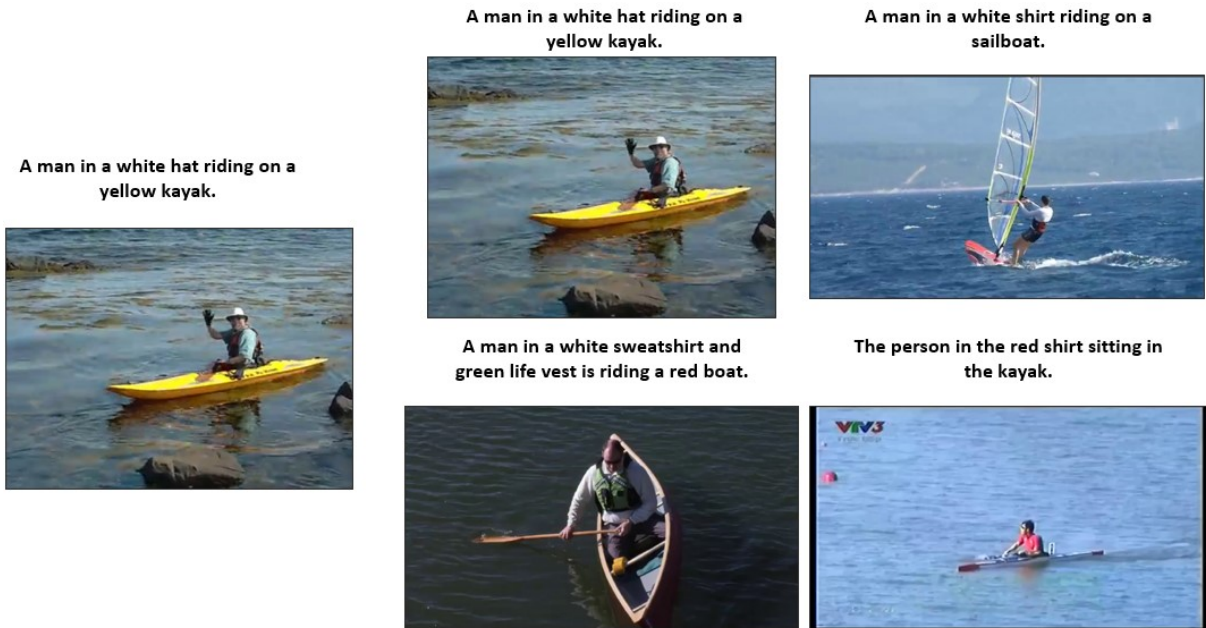
Source: Author.

an example of the results obtained in the execution of the model focused on a 1st, 2nd, and 3rd match, as well as an example where there was no match in any of the first four ranked videos, is presented to have a deeper understanding of the model's behavior. The analysis is focused on the combination of the scenes, subjects, and actions taking place, as identified in the sentences.

Figure 28 successfully returned the expected video at the first position, but looking at the other top-ranked images, it is possible to observe that all images are related to water sports, more specifically, boat-related being conducted by a man. The color of the boat was not matched in the subsequent images, nor was the vest of the person, understanding that it took into account the scene and the main information from the NL sentence.

In Figure 29 the model was able to identify the action of mixing a drink behind a bar, but not in the last one, in which a desk can be seen as some sort of bar, where the person is located behind it. The fact that only one of the videos is about a woman suggests that the person's gender was not obvious. The test dataset was analyzed to make sure there were other video

Figure 28 – Retrieval example matching the 1st ranked image.



Source: Author.

Figure 29 – Retrieval example matching the 2nd ranked image.



Source: Author.

examples of a woman making drinks. In the test dataset, there are eight drink-related videos, of which three have a woman mixing them.

The ranked sequence of Figure 30 presented a good relationship between the expected and the presented videos, even because the ground truth image was located at the 3rd position, only. In all the videos, a person is playing the guitar. Also, in all of them, the person is wearing

Figure 30 – Retrieval example matching the 3rd ranked image.



Source: Author.

dark clothes.

Figure 31 – Retrieval example with no match on ranked image.



Source: Author.

Finally, analyzing Figure 31, the example in which there was no matching ranked video with the ground truth one, it is possible to see that in all of them there are an animal and a person in an outside space. Of the four images, three are dog-related, as expected, and in one of the dog-related videos, the person is playing frisbee, as happened in the ground truth video. It is

possible to infer that the model understood the scene and the animal relationship, failing to find the correct video although returning relevant videos.

4.6.1 Ad hoc analysis

The previous analysis was done over existing sentences and videos, where it is possible to compare what was expected with what was retrieved. This Section presents the results obtained when executing the retrieval process over sentences that are not part of the test dataset.

There are two approaches performed: (1) using random sentences, not based on any visual content; and (2) using sentences generated after watching available videos in the test dataset, knowing what should be returned.

Figure 32 shows the retrieved images after a round of random sentences, while Figure 33 shows the retrieved images after a round of observed videos.

It is easily noticed that the random free sentences did not meet the expectations for retrieval tasks. The first clear observation is that the model does not perform well on small sentences or atomic words. Simply entering a word like “woman” returned random videos with no relation at all to what would be expected. This is probably because the model was not trained for single-word recognition, needing longer sentences to generate the vector that it was trained to compare.

The second observation is to note that the quality of the retrieval is related to the available examples on the test dataset. When there are many videos related to the topic, the retrieval performs better. But, when there is only a single video about the topic, it does not work well. It is possible to infer that the more video examples a specific topic may have, the easier it is for the model to identify it.

When observing the results from the retrieval using sentences from video observation, it is possible to notice that the results are better and closer to what was searched for. The subject being queried is visible in most cases. The other related videos also exhibit similar scenes (gymnastic equipment, instruments, etc).

The improvement in the results may be related to both (1) the situation where the sentence is directly related to a video that is available and (2) the improvement in the description of the sentence.

Given the tests done and the observations of the results, it is not possible to answer Question 11 by affirming that the model bases itself only on an action or on a scene, but it seems

to have a balanced distribution of importance between both. Querying specific actions may return videos that are related to them (“dancing”, for example), but it may also return videos where the scene is similar, like indoor situations or where a wooden background can be seen, or having similar objects on the scene, like “bars” or “animals”.

Also, it is possible to say that the model gives high importance to the verb, or action, relating actions such as “riding” and “playing” with videos that have semantic similarities. It is also possible to notice that the nouns also represent an important part of the query. That is easy to notice when comparing results from queries such as “playing an instrument” and “playing a guitar” because both will return musical instruments, but the second will target a specific instrument. That answers Question 12.

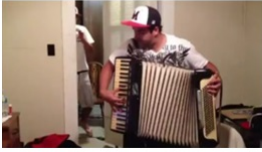










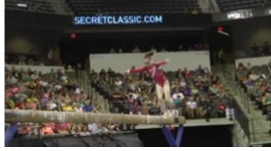




Finally, querying words that are not part of the corpus, like “Easter” or “mainframe” returns results based on the rest of the sentence query because those unknown words are ignored by the model, answering Question 13.

Figure 32 – Retrieval example for random free sentences.

<p>A woman playing an instrument.</p>	<p>A dark haired man with white headphones is playing a violin.</p>	<p>A man with glasses and white striped shirt is playing the flute in front of a microphone.</p>
		
	<p>A woman with dark short hair playing on a flute.</p>	<p>A girl in black and white plays the guitar.</p>
		
<p>A man is riding a boat.</p>	<p>A man in a white shirt riding on a sailboat.</p>	<p>A man in a white hat riding on a yellow kayak.</p>
		
	<p>A man in a white sweatshirt and green life vest is riding in a red boat.</p>	<p>A man in a blue kayak riding in the rapids.</p>
		
<p>Man in black.</p>	<p>A person in a yellow shirt practicing the hammer throw.</p>	<p>A man in orange shirt playing racketball.</p>
		
	<p>A man in a white shirt is spinning and throwing a ball.</p>	<p>A man in white is performing tai chi moves.</p>
		
<p>A man in black is riding a bicycle.</p>	<p>A person in a wet-suit is surfing in the ocean.</p>	<p>A man in a black wetsuit surfing on a wooden board.</p>
		
	<p>A man in a white shirt riding on a sailboat.</p>	<p>A man in a black jacket and white helmet riding a bike.</p>
		

Source: Author.

Figure 33 – Retrieval example for video observation free sentences.

A man playing drums.	A man in a white hat and black shirt playing the accordion.		A man is playing drums on a stage.	
	A toddler wearing white clothes inside a house bangs on some drums with her hands.		A man in orange shirt is drumming.	
A person lifting weight.	A man in red is lifting a heavy weight.		A man in blue shorts is sparring with another person.	
	A man in dark blue is coaching another man in a gym.		A boy in white pants swinging on bars in a gym.	
A person performing gymnastics.	A man in red is lifting a heavy weight.		A man in blue shorts is sparring with another person.	
	A gymnast wearing a blue and white leotard and a ponytail is performing a routine on the balance beam.		A gymnast in red white and blue is on the balance beam.	
A woman dancing.	A girl in a tutu is twirling a baton.		A woman in a blue bikini top and skirt dances.	
	A man in a red and white uniform is lifting a weightlifting bar with large black weights on each end.		A person in a white jacket and black pants is fencing.	

Source: Author.

5 CONCLUSIONS AND FUTURE WORKS

This Chapter presents the contributions of this dissertation for the video retrieval from a NL query. The achievements and the limitations of the proposed work are discussed according to the experiments done. Recommendations for improvements and future works are also presented.

In this work, we introduced a study related to how to retrieve a video, from a list of available videos, using a NL query. A model using a triplet siamese network was presented with the evolution of its configuration and parameters as well as the impact of specific models for feature extraction.

The initial tests aimed to achieve the best model architecture for the work. Stacked RNN layers showed a negative impact on the learning capability of the models, probably related to the excessive abstraction level they created. On the other hand, a stacked sequence of dense layers presented an improvement in the learning of the models. Different optimizers were also tested, showing that faster optimizers, like RMSProp and Adam, led to a quicker convergence, causing overfitting, while a slower optimizer, like SGD, led the model to a more stable learning rate. It was possible to understand that the size of the available information resultant from both video and NLP extraction models had a direct impact on the learning, suggesting that cutting or reducing the size of the vectors may harm the learning.

Next, four different CNN models for video feature extraction were tested to show the differences of VGG16, ResNet50, InceptionV3 and MobileNetV2. From these models, MobileNetV2 achieved the best performance, leading to an understanding that newer models are more effective in extracting features, possibly having a smarter or better-designed internal structure, in which the most meaningful features are taken into consideration, even having a smaller number of trainable parameters.

On the PoS impact, a test comparing the results when using only *nouns* and *adjectives* were compared with when using those plus *verbs* and *adverbs*. Even though the difference in the maximum size of available tokens was not aggressive, it was possible to observe that the usage of *verbs* for action descriptions led the model to better understand what was being requested in the query, thus resulting in better results.

In the tests related to the embedding space, the impact of the α value used to fine-tune the loss function of the triplet architecture was tested. It was clear that a higher α negatively impacted the learning rate. Probably, this was caused by removing the subtle capability of small

adjustments on the loss function, similarly to what a small α did, due to not punishing negative examples as expected.

The controlled selection of specific negative examples for a given anchor showed that the distance to their examples is of high importance to the quality of the training. If the distance from the video to the anchor is too high or too low, the model may be negatively affected, leading to bad results. We observed that there is an optimal range for the distance, between 25% to 50% close to the anchor. Yet, the use of a dataset of negative examples created by a random selection returned better results, probably because it was created with a mixture of good and bad examples, increasing the robustness of the dataset.

The observation on the *ad hoc* queries using free texts showed that the models do not understand single words, or atomic, queries. Simply querying words like “man” or “woman” returns almost random results. Improved queries return much better results, but they appear to be directly related to the number of available video examples and have difficulty retrieving actions with few examples. Besides, the model gives high importance to the verb, or action, related to the video, followed by considering the similar scene surroundings. The importance of nouns was also significant, but it was directly related to the number of example videos used during tests. “Instruments”, “animals”, “boats” and other common nouns presented good results, while not-so-common nouns, such as “hammer” had poor results. Adjectives were not highly effective in retrieving the videos, probably because there are many different adjectives for the same noun, making it harder to train specific words such as colors.

This work used a batch approach for all of its feature extraction processes. That approach was selected because the real-time response does not matter and does not affect the results. On the one hand, the batch approach requires a large amount of storage available to store the preprocessed videos and sentences, taking a long processing time before the learning and testing phases start. On the other hand, it makes the tests faster, given the fact that all the required infrastructure requirements are already available. The online approach aims for the opposite advantages and disadvantages, not needing any kind of preprocessing steps as well as large storage availability. However, it does require a long training and testing time, since it requires all feature extractions to occur simultaneously with the learning and testing.

Overall, the results obtained in this work were promising. A possible list of future work may include the exploration of an online processing approach to understand the model’s behavior and its impact on the response time and quality of results. Also, there are newer

CNN models available for video extraction that could be tested, as well as test models that take advantage of a single layer, such as C3D for video and BERT for NLP. As identified in Section 2.2.2.1, item “c) Embedding space-based solutions”, there is a new line of study that uses multi-modal Transformers to enrich the retrieval using several different modalities available in the same video and that should be further explored. Regarding the controlled selection of negative examples, a combined approach where the videos are randomly selected but from a specific and controlled range of videos may improve the training dataset. Finally, training the model with atomic examples may lead to better learning of the model to be more generalist when queried with less complex sentences.

REFERENCES

ABADI, Martín; AGARWAL, Ashish; BARHAM, Paul *et al.* **TensorFlow: large-scale machine learning on heterogeneous systems**. 2015. Software available from tensorflow.org. Available at: <http://tensorflow.org/>.

AKULA, Jayaprakash; DABRAL, Rishabh; JYOTHI, Preethi; RAMAKRISHNAN, Ganesh. Cross lingual video and text retrieval: a new benchmark dataset and algorithm. *In: Proc. of International Conference on Multimodal Interaction*. [S.l.: s.n.], 2021. p. 595–603.

BANSAL, Ravi; CHAKRABORTY, Sandip. Visual content based video retrieval on natural language queries. *In: Proc. of 34th ACM/SIGAPP Symposium on Applied Computing*. [S.l.: s.n.], 2019. p. 212–219.

BARRETT, Daniel Paul; BARBU, Andrei; SIDDHARTH, N; SISKIND, Jeffrey Mark. Saying what you're looking for: Linguistics meets video search. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 10, p. 2069–2081, 2015.

BEZEMER, Jeff; JEWITT, Carey. Multimodality: A guide for linguists. **Research Methods in Linguistics**, v. 28, p. 1–3, 2018.

BRADSKI, G. The opencv library. **Dr. Dobb's Journal of Software Tools**, 2000.

CAO, Da; ZENG, Yawen; WEI, Xiaochi; NIE, Liqiang; HONG, Richang; QIN, Zheng. Adversarial video moment retrieval by jointly modeling ranking and localization. *In: Proc. of 28th ACM International Conference on Multimedia*. [S.l.: s.n.], 2020. p. 898–906.

CARREIRA, João; ZISSERMAN, Andrew. Quo vadis, action recognition? a new model and the kinetics dataset. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 6299–6308.

CHEN, Cheng; GU, Xiaodong. Context-aware network with foreground recalibration for grounding natural language in video. **Neural Computing and Applications**, v. 33, n. 16, p. 10485–10502, 2021.

CHEN, Qingchao; LIU, Yang; ALBANIE, Samuel. Mind-the-gap! unsupervised domain adaptation for text-video retrieval. *In: Proc. of AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2021. v. 35, n. 2, p. 1072–1080.

CHEN, Yahui. **Convolutional Neural Network for Sentence Classification**. 2015. Master's Thesis (Master of Mathematics in Computer Science) — University of Waterloo, Canada, 2015.

CHO, Kyunghyun; MERRIËNBOER, Bart Van; BAHDANAU, Dzmitry; BENGIO, Yoshua. On the properties of neural machine translation: encoder-decoder approaches. **arXiv preprint arXiv:1409.1259**, 2014.

CHO, Kyunghyun; MERRIËNBOER, Bart Van; GULCEHRE, Caglar; BAHDANAU, Dzmitry; BOUGARES, Fethi; SCHWENK, Holger; BENGIO, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.

CHOLLET, Francois *et al.* **Keras**. GitHub, 2015. Available at: <https://github.com/fchollet/keras>.

COTTER, Kelley. Playing the visibility game: How digital influencers and algorithms negotiate influence on instagram. **New Media & Society**, v. 21, n. 4, p. 895–913, 2019.

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DONG, Jianfeng; LI, Xirong; XU, Chaoxi; JI, Shouling; HE, Yuan; YANG, Gang; WANG, Xun. Dual encoding for zero-example video retrieval. *In: Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 9346–9355.

DONG, Jianfeng; LI, Xirong; XU, Chaoxi; YANG, Xun; YANG, Gang; WANG, Xun; WANG, Meng. Dual encoding for video retrieval by text. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 8, p. 4065–4080, 2021.

ENSER, Peter; SANDOM, Christine. Towards a comprehensive survey of the semantic gap in visual image retrieval. *In: Proc. of International Conference on Image and Video Retrieval*. [S.l.: s.n.], 2003. p. 291–299.

ESCORCIA, Victor; HEILBRON, Fabian Caba; NIEBLES, Juan Carlos; GHANEM, Bernard. Daps: Deep action proposals for action understanding. *In: Proc. European Conference on Computer Vision*. [S.l.: s.n.], 2016. p. 768–784.

FAGHRI, Fartash; FLEET, David J; KIROS, Jamie Ryan; FIDLER, Sanja. Vse++: Improving visual-semantic embeddings with hard negatives. **arXiv preprint arXiv:1707.05612**, 2017.

FAN, Hehe; YANG, Yi. Person tube retrieval via language description. *In: Proc. of AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2020. v. 34, n. 07, p. 10754–10761.

GABEUR, Valentin; SUN, Chen; ALAHARI, Karteek; SCHMID, Cordelia. Multi-modal transformer for video retrieval. *In: Proc. European Conference on Computer Vision*. [S.l.: s.n.], 2020. p. 214–229.

GAO, Jiyang; SUN, Chen; YANG, Zhenheng; NEVATIA, Ram. Tall: temporal activity localization via language query. *In: Proc. of IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 5267–5275.

GUTOSKI, M.; RIBEIRO, M.; HATTORI, L. T.; ROMERO, M.; LAZZARETTI, A. E.; LOPES, H. S. A comparative study of transfer learning approaches for video anomaly detection. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 35, n. 5, p. 2152003, 2021.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 770–778.

HEILBRON, Fabian Caba; ESCORCIA, Victor; GHANEM, Bernard; NIEBLES, Juan Carlos. Activitynet: a large-scale video benchmark for human activity understanding. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 961–970.

HENDRICKS, Lisa Anne; WANG, Oliver; SHECHTMAN, Eli; SIVIC, Josef; DARRELL, Trevor; RUSSELL, Bryan. Localizing moments in video with natural language. *In: Proc. of IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 5803–5812.

HENDRICKS, Lisa Anne; WANG, Oliver; SHECHTMAN, Eli; SIVIC, Josef; DARRELL, Trevor; RUSSELL, Bryan. Localizing moments in video with temporal language. *arXiv preprint arXiv:1809.01337*, 2018.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.

HOU, Zhijian; NGO, Chong-Wah; CHAN, Wing Kwong. CONQUER: contextual query-aware ranking for video corpus moment retrieval. *In: Proc. of 29th ACM International Conference on Multimedia*. [S.l.: s.n.], 2021. p. 3900–3908.

JAIMES, Alejandro; SEBE, Nicu. Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding*, v. 108, n. 1-2, p. 116–134, 2007.

JIANG, Siyu; WU, Guobin. Mrn: moment relation network for natural language video localization with transfer learning. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 35, n. 7, p. 2152009, 2021.

KIPF, Thomas N; WELLING, Max. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

KITCHENHAM, Barbara; PRETORIUS, Rialette; BUDGEN, David; BRERETON, O Pearl; TURNER, Mark; NIAZI, Mahmood; LINKMAN, Stephen. Systematic literature reviews in software engineering – a tertiary study. **Information and Software Technology**, v. 52, n. 8, p. 792–805, 2010.

KLEIN, Dan; MANNING, Christopher D. Accurate unlexicalized parsing. *In: Proc. of 41st Annual Meeting of the Association for Computational Linguistics. [S.l.: s.n.]*, 2003. p. 423–430.

LI, Ding; WU, Rui; TANG, Yongqiang; ZHANG, Zhizhong; ZHANG, Wensheng. Multi-scale 2D representation learning for weakly-supervised moment retrieval. *In: Proc. of 25th International Conference on Pattern Recognition. [S.l.: s.n.]*, 2021. p. 8616–8623.

LIU, Bingbin; YEUNG, Serena; CHOU, Edward; HUANG, De-An; FEI-FEI, Li; NIEBLES, Juan Carlos. Temporal modular networks for retrieving complex compositional activities in videos. *In: Proc. of European Conference on Computer Vision. [S.l.: s.n.]*, 2018. p. 552–568.

LIU, Meng; WANG, Xiang; NIE, Liqiang; HE, Xiangnan; CHEN, Baoquan; CHUA, Tat-Seng. Attentive moment retrieval in videos. *In: Proc. of 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. [S.l.: s.n.]*, 2018. p. 15–24.

LIU, Meng; WANG, Xiang; NIE, Liqiang; TIAN, Qi; CHEN, Baoquan; CHUA, Tat-Seng. Cross-modal moment localization in videos. *In: Proc. of 26th ACM International Conference on Multimedia. [S.l.: s.n.]*, 2018. p. 843–851.

LIU, Xinfang; NIE, Xiushan; TENG, Junya; LIAN, Li; YIN, Yilong. Single-shot semantic matching network for moment localization in videos. **ACM Transactions on Multimedia Computing, Communications, and Applications**, v. 17, n. 3, p. 1–14, 2021.

LIU, Yang; ALBANIE, Samuel; NAGRANI, Arsha; ZISSERMAN, Andrew. Use what you have: Video retrieval using representations from collaborative experts. **arXiv preprint arXiv:1907.13487**, 2019.

LOPER, Edward; BIRD, Steven. NLTK: the natural language toolkit. *In: Proc. of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. [S.l.: s.n.]*, 2002.

MA, Minuk; YOON, Sunjae; KIM, Junyeong; LEE, Youngjoon; KANG, Sunghun; YOO, Chang D. Vlanet: video-language alignment network for weakly-supervised video moment retrieval. *In: Proc. European Conference on Computer Vision. [S.l.: s.n.]*, 2020. p. 156–171.

McCulloch, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, 1943.

- MIECH, Antoine; ZHUKOV, Dimitri; ALAYRAC, Jean-Baptiste; TAPASWI, Makarand; LAPTEV, Ivan; SIVIC, Josef. Howto100m: learning a text-video embedding by watching hundred million narrated video clips. *In: Proc. of IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 2630–2640.
- Mohsin, Maryam. **10 Youtube Statistics That You Need to Know in 2021**. [S.l.]: Oberlo, 2021. <https://www.oberlo.com/blog/youtube-statistics>. [Online; accessed 2021-07-25].
- OTANI, Mayu; NAKASHIMA, Yuta; RAHTU, Esa; HEIKKILÄ, Janne; YOKOYA, Naokazu. Learning joint representations of videos and sentences with web image search. *In: Proc. European Conference on Computer Vision*. [S.l.: s.n.], 2016. p. 651–667.
- PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. Glove: global vectors for word representation. *In: Proc. of Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2014. p. 1532–1543.
- QI, Mengshi; QIN, Jie; YANG, Yi; WANG, Yunhong; LUO, Jiebo. Semantics-aware spatial-temporal binaries for cross-modal video retrieval. **IEEE Transactions on Image Processing**, v. 30, p. 2989–3004, 2021.
- QU, Xiaoye; TANG, Pengwei; ZOU, Zhikang; CHENG, Yu; DONG, Jianfeng; ZHOU, Pan; XU, Zichuan. Fine-grained iterative attention network for temporal language localization in videos. *In: Proc. of 28th ACM International Conference on Multimedia*. [S.l.: s.n.], 2020. p. 4280–4288.
- REGNERI, Michaela; ROHRBACH, Marcus; WETZEL, Dominikus; THATER, Stefan; SCHIELE, Bernt; PINKAL, Manfred. Grounding action descriptions in videos. **Transactions of the Association for Computational Linguistics**, v. 1, p. 25–36, 2013.
- REIMERS, Nils; GUREVYCH, Iryna. Sentence-BERT: sentence embeddings using siamese BERT-networks. *In: Proc. of Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2019.
- SAH, Shagan; GOPALAKISHNAN, Sabarish; PTUCHA, Raymond. Aligned attention for common multimodal embeddings. **Journal of Electronic Imaging**, v. 29, n. 2, p. 023013, 2020.
- SANDLER, Mark; HOWARD, Andrew; ZHU, Menglong; ZHMOGINOV, Andrey; CHEN, Liang-Chieh. Mobilenetv2: inverted residuals and linear bottlenecks. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 4510–4520.
- SCARSELLI, Franco; GORI, Marco; TSOI, Ah Chung; HAGENBUCHNER, Markus; MONFARDINI, Gabriele. The graph neural network model. **IEEE Transactions on Neural Networks**, v. 20, n. 1, p. 61–80, 2008.

SCHROFF, Florian; KALENICHENKO, Dmitry; PHILBIN, James. Facenet: a unified embedding for face recognition and clustering. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.]*, 2015. p. 815–823.

SHAO, Dian; XIONG, Yu; ZHAO, Yue; HUANG, Qingqiu; QIAO, Yu; LIN, Dahua. Find and focus: retrieve and localize video events with natural language queries. *In: Proc. of European Conference on Computer Vision. [S.l.: s.n.]*, 2018. p. 200–216.

SIGURDSSON, Gunnar A; VAROL, Gül; WANG, Xiaolong; FARHADI, Ali; LAPTEV, Ivan; GUPTA, Abhinav. Hollywood in homes: crowdsourcing data collection for activity understanding. *In: Proc. of European Conference on Computer Vision. [S.l.: s.n.]*, 2016. p. 510–526.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SUN, Xiaoyang; WANG, Hanli; HE, Bin. MABAN: multi-agent boundary-aware network for natural language moment retrieval. **IEEE Transactions on Image Processing**, v. 30, p. 5589–5599, 2021.

SZEGEDY, Christian; VANHOUCKE, Vincent; IOFFE, Sergey; SHLENS, Jon; WOJNA, Zbigniew. Rethinking the inception architecture for computer vision. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.]*, 2016. p. 2818–2826.

TAI, Kai Sheng; SOCHER, Richard; MANNING, Christopher D. Improved semantic representations from tree-structured long short-term memory networks. **arXiv preprint arXiv:1503.00075**, 2015.

TANG, Haoyu; ZHU, Jihua; LIU, Meng; GAO, Zan; CHENG, Zhiyong. Frame-wise cross-modal matching for video moment retrieval. **IEEE Transactions on Multimedia**, v. 24, p. 1338–1349, 2022.

TANG, Kevin; YAO, Bangpeng; FEI-FEI, Li; KOLLER, Daphne. Combining the right features for complex event recognition. *In: Proc. of IEEE International Conference on Computer Vision. [S.l.: s.n.]*, 2013. p. 2696–2703.

TOMAR, Suramya. Converting video formats with FFmpeg. **Linux Journal**, v. 2006, n. 146, p. 10, 2006.

VORDERER, Peter; KLIMMT, Christoph. The mobile user’s mindset in a permanently online, permanently connected society. **The Oxford Handbook of Mobile Communication and Society**, Oxford University Press, p. 54, 2020.

WANG, Hao; ZHA, Zheng-Jun; CHEN, Xuejin; XIONG, Zhiwei; LUO, Jiebo. Dual path interaction network for video moment localization. *In: Proc. of 28th ACM International Conference on Multimedia*. [S.l.: s.n.], 2020. p. 4116–4124.

WANG, Wei; GAO, Junyu; YANG, Xiaoshan; XU, Changsheng. Learning coarse-to-fine graph neural networks for video-text retrieval. **IEEE Transactions on Multimedia**, v. 23, p. 2386–2397, 2020.

WANG, Xiaohan; ZHU, Linchao; YANG, Yi. T2vlad: global-local sequence alignment for text-video retrieval. *In: Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 5079–5088.

WRAY, Michael; LARLUS, Diane; CSURKA, Gabriela; DAMEN, Dima. Fine-grained action retrieval through multiple parts-of-speech embeddings. *In: Proc. of IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 450–459.

WU, Jie; LI, Guanbin; HAN, Xiaoguang; LIN, Liang. Reinforcement learning for weakly supervised temporal grounding of natural language in untrimmed videos. *In: Proc. of 28th ACM International Conference on Multimedia*. [S.l.: s.n.], 2020. p. 1283–1291.

XU, Jun; MEI, Tao; YAO, Ting; RUI, Yong. MSR-VTT: a large video description dataset for bridging video and language. *In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 5288–5296.

YAMAGUCHI, Masataka; SAITO, Kuniaki; USHIKU, Yoshitaka; HARADA, Tatsuya. Spatio-temporal person retrieval via natural language queries. *In: Proc. of IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 1453–1462.

YANG, Xun; DONG, Jianfeng; CAO, Yixin; WANG, Xun; WANG, Meng; CHUA, Tat-Seng. Tree-augmented cross-modal encoding for complex-query video retrieval. *In: Proc. of 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.: s.n.], 2020. p. 1339–1348.

YU, Xinyan; ZHANG, Ya; ZHANG, Rui. Cross-modality video segment retrieval with ensemble learning. *In: SINGH, R.; VATSA, M.; PATEL, V.; RATHA, N. (Ed.). Domain Adaptation for Visual Understanding*. Cham: Springer, 2020. p. 65–79.

YU, Youngjae; KIM, Jongseok; KIM, Gunhee. A joint sequence fusion model for video question answering and retrieval. *In: Proc. of European Conference on Computer Vision*. [S.l.: s.n.], 2018. p. 471–487.

ZHANG, Da; DAI, Xiyang; WANG, Xin; WANG, Yuan-Fang; DAVIS, Larry S. MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. *In:*

Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. [*S.l.: s.n.*], 2019. p. 1247–1257.

ZHANG, Pengju; YUAN, Chunmiao; LIU, Kunliang; SUN, Yukuan; LIANG, Jiayu; JIN, Guanghao; WANG, Jianming. Fast video clip retrieval method via language query. *In: Proc. of International Conference on Advanced Data Mining and Applications.* [*S.l.: s.n.*], 2019. p. 526–534.

ZHANG, Songyang; SU, Jinsong; LUO, Jiebo. Exploiting temporal relationships in video moment localization with natural language. *In: Proc. of 27th ACM International Conference on Multimedia.* [*S.l.: s.n.*], 2019. p. 1230–1238.

ZHANG, Zhu; LIN, Zhijie; ZHAO, Zhou; XIAO, Zhenxin. Cross-modal interaction networks for query-based moment retrieval in videos. *In: Proc. of 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* [*S.l.: s.n.*], 2019. p. 655–664.