Exploring and Visualizing Referring Expression Comprehension

Bachelor's Thesis (Mathematics & Industrial Engineering)

David Álvarez Rosa[1,2,3]

Supervisor: Sanja Fidler[5] *Co-Supervisor:* Xavier Giró[4]

Politechnical University of Catalonia
[1]Interdisciplinary Higher Education Centre
[2]Barcelona School of Industrial Engineering — [3]School of Mathematics and Statistics
[4]Signal Theory and Communications Department

University of Toronto
[5]Computer Science Department

Barcelona – May 25, 2021

Table of Contents

1. Chapter 1. Introduction
2. Chapter 2. Theoretical Background
3. Chapter 3. Referring Expression Comprehension
4. Chapter 4. Models
5. Chapter 5. Results and Comparison
6. Chapter 6. Visualization
7. Chapter 7. Project Analysis

Chapter Outline

1. Chapter 1. Introduction
   - Description and Motivation
   - Applications

Problem Description

Referring Expression Comprehension

Image + Referring Expression ⟶ Segmentation!

(a) Man with cap (b) Laptop on the right (c) Army officer white suit

Figure: Examples of Referring Expression Comprehension

Problem Description

Referring Expression Comprehension

Image + Referring Expression ⟶ Segmentation!

(a) Man with cap (b) Laptop on the right (c) Army officer white suit

Figure: Examples of Referring Expression Comprehension

Problem Description

Referring Expression Comprehension

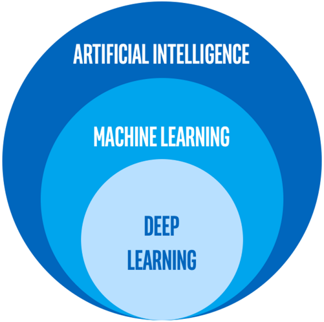
Image + Referring Expression $\longrightarrow$ Segmentation!

(a) Man with cap   (b) Laptop on the right   (c) Army officer white suit

Figure: Examples of Referring Expression Comprehension

Big Frame

Referring Expression Comprehension



Regarding learning techniques

- Artificial Intelligence
- Machine Learning
- Deep Learning

Regarding type of data

- Computer Vision
- Natural Language Processing
- . . . and Multimodal Learning

Big Frame

Referring Expression Comprehension

Regarding learning techniques

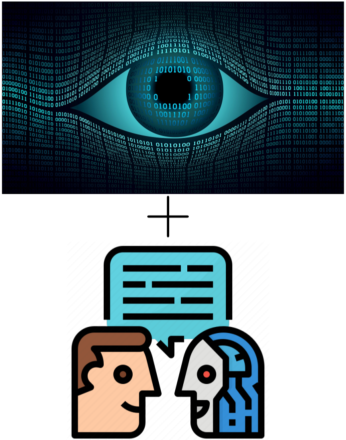
- Artificial Intelligence
- Machine Learning
- Deep Learning

Regarding type of data

- Computer Vision
- Natural Language Processing
- . . . and Multimodal Learning

Big Frame

Referring Expression Comprehension



Regarding learning techniques

- Artificial Intelligence
- Machine Learning
- Deep Learning

Regarding type of data

- Computer Vision
- Natural Language Processing
- . . . and Multimodal Learning

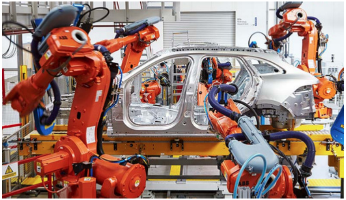
Objectives

Bachelor's Thesis

- Learning about Machine Learning (ML)
- Fundamentals of neural models
- Undestand state-of-the-art papers
- Model (in REC and Speech to Text)
- Front end development (HTML, CSS, JS)
- Back end (PHP)
- Academia (creationg of thesis and presentation)

Applications

Referring Expression Comprehension



- Theoretical
- Industry
- Home Automation and IoT
- Security

Chapter Outline

2. Chapter 2. Theoretical Background
   - Tensors
   - Neural Network Architectures
   - Training
   - Testing

Machine Learning Overview

From Reality to Fiction

Mathematical perspective

Given dataset Ω of inputs $x \in \mathbb{R}^n$ and outputs $y \in \mathbb{R}^m$, find

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ x &\longmapsto f(x) := \hat{y}, \end{aligned} \tag{1}$$

such that $\hat{y} \approx y$ for every element in the dataset.

Desired generalization

We do not seek memorization, we seek to extract relevant information from the structure of the data in order to make predictions.

Machine Learning Overview

From Reality to Fiction

Mathematical perspective

Given dataset Ω of inputs $x \in \mathbb{R}^n$ and outputs $y \in \mathbb{R}^m$, find

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ x &\longmapsto f(x) := \hat{y}, \end{aligned} \tag{1}$$

such that $\hat{y} \approx y$ for every element in the dataset.

Desired generalization

We do not seek memorization, we seek to extract relevant information from the structure of the data in order to make predictions.

Tensors

View as multidimensional arrays

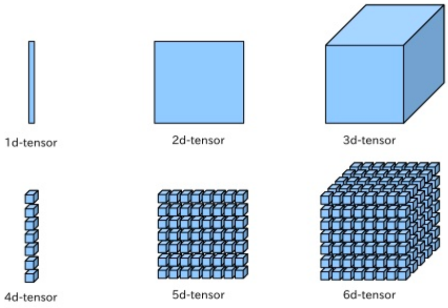


Figure: Tensor representation

- We will understand the tensors as a grouping data structure,
- i.e, as multidimensional arrays.
- T_i, with index $i = (i_1, \dots, i_n)$

Tensor example

Images as tensor of rank 3,

$$I \in \mathbb{R}^{C \times H \times W}. \tag{2}$$

Neural Network Architectures

Overview

- Feedforward Neural Network
- Convolutional Neural Network
- Recurrent Neural Network
- Transformer Model

Feedforward Neural Network

Topology: Layers, Neurons and Connections

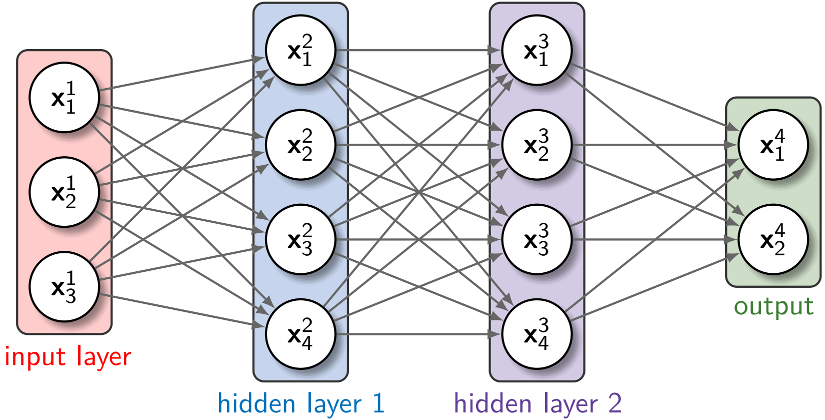


Figure: Feedforward Neural Network architecture

Feedforward Neural Network

Mathematical Representation

Forward computation

The output values are $y = x^L$ can be computed recursively,

$$x^{\ell+1} = f\left(W^{\ell}x^{\ell} + b^{\ell}\right), \tag{3}$$

where W^l and b^l are a matrix and a vector of weights respectively.

Example of digit recognition

Convolutional Neural Network

Topology

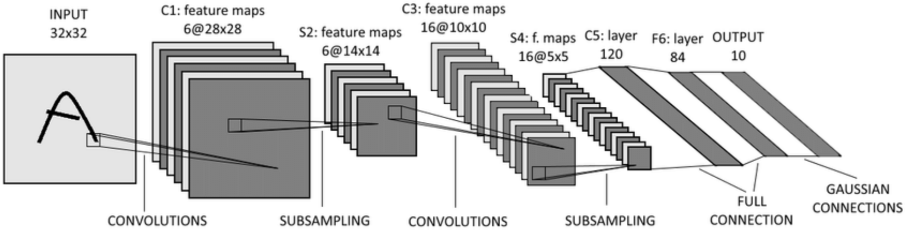


Figure: Convolutional Neural Network architecture

Convolutional Neural Network

Convolutional Layer

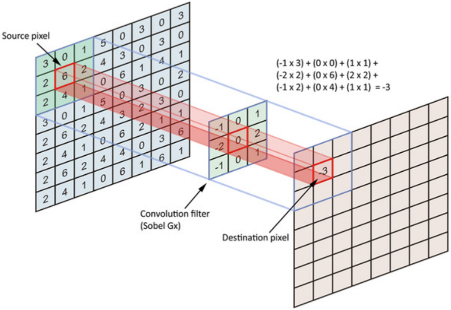


Figure: Convolution

Mathematical definition

Given filter F, the convolution is defined as follows,

$$Y_{i,j,k} = \sum_{l,m,n} X_{l,j+m,k+n} F_{i,l,m,n}, \quad (4)$$

where the sum is performed for all valid l, m, n indices.

Convolutional Neural Network

Pooling Layer

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

$\xrightarrow{2 \times 2 \text{ Max-Pool}}$

20	30
112	37

Figure: Max pooling

- Reduce network dimension
- Add non-linearities
- Enlarge field of view

Max pooling

$$f_{X,Y}(S) = \max_{a,b=0}^{1} S_{2X+a,2Y+b} \tag{5}$$

Recurrent Neural Network

Topology

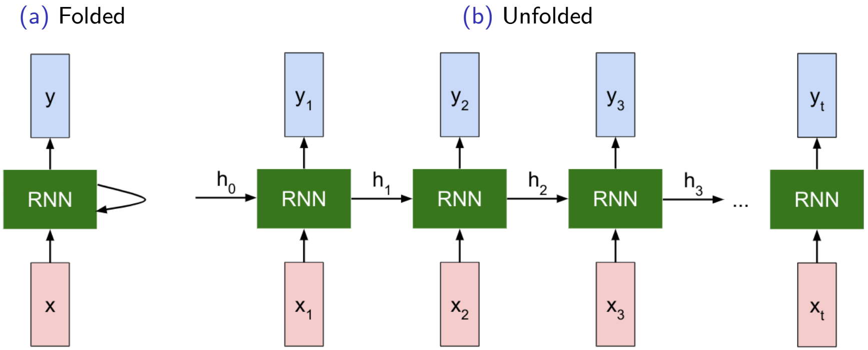


Figure: Recurrent Neural Netowrk architecture

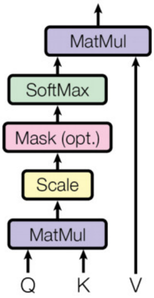
Recurrent Neural Network

Variants

- Long Short Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- But, . . .

Transformer Model

Topology



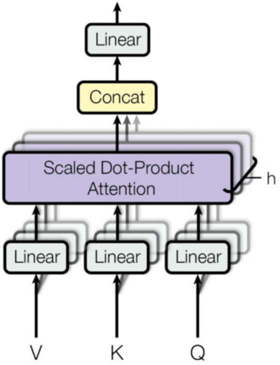
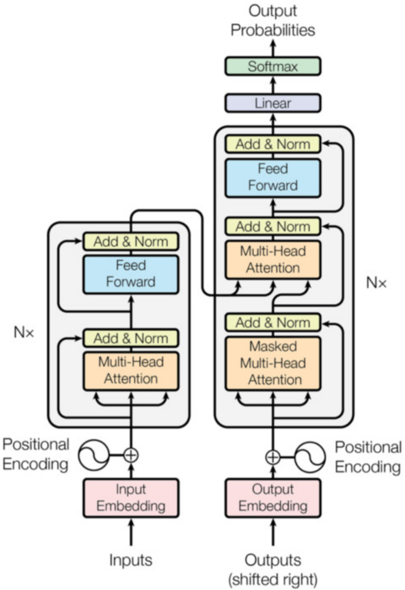


Figure: Attention mechanism


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

Training overview

Loss functions

- Measures differences between desired and predicted
- Dataset $\Omega = \{(x_i, y_i)\}_{i=1}^{N}$
- And, predicted output $\hat{y} = f(x)$
- It is common that,

$$\mathcal{L}(\Omega, \theta) = \frac{1}{N} \sum_{(x,y)\in\Omega} \ell(x, y, \theta). \tag{6}$$

- We will assume $\ell \in \mathcal{C}^1$

Optimization

Overall idea

- We will seek to approximate,

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\Omega, \theta), \tag{7}$$

given that the above exists.

- Usually iterative methods, i.e., given initial guess $\theta^{(0)}$, proceed as follows,

$$\theta^{(t+1)} = \theta^{(t)} + \alpha\, \Delta\theta^{(t)}, \tag{8}$$

where α is called the step size (or learning rate), and $\Delta\theta^{(t)}$ is the weight update in step t.

Optimization

Methods

- First order methods (higher order infeasible)
- Gradient Descent:

$$\Delta\theta^{(t)} = -\nabla\mathcal{L}(\theta^{(t)}). \tag{9}$$

  Due to loss function,

$$\nabla_\theta \mathcal{L}(\Omega, \theta) = \frac{1}{N} \sum_{(x,y)\in\Omega} \nabla_\theta \ell(x, y, \theta). \tag{10}$$

- Therefore, in practice, Stochastic Gradient Descent,

$$\nabla_\theta \mathcal{L}(\Omega, \theta) \approx \frac{1}{|B|} \sum_{(x,y)\in B} \nabla_\theta \ell(x, y, \theta). \tag{11}$$

Optimization

Methods II

Avoid saddle points

Using momentum,

$$\Delta\theta^{(t)} = -\beta\Delta\theta^{(t-1)} - \nabla\mathcal{L}(\theta^{(t)}), \tag{12}$$

where β is the “friction” hyperparameter.

Optimization

Weight initialization

- Random
- Xavier initialization

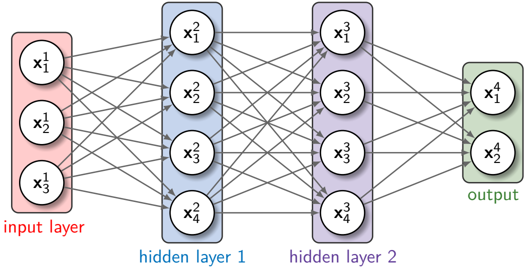
Optimization

Computing derivatives: error backpropagation

Backpropagation algorithm exemplified

Let w_{ij}^l be an individual weight, involved in computing $\hat{y}_i^l$ then the chain rule applied to $\ell(x, y, \theta)$ yields,

$$\frac{\partial \ell}{\partial w_{ij}^l} = \frac{\partial \ell}{\partial \hat{y}_i^l} \frac{\partial y_i^l}{\partial w_{ij}^l}. \tag{13}$$



Regularization Techniques

Understanding overfitting

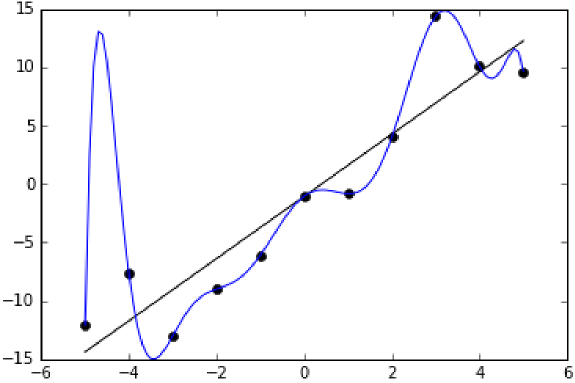


Figure: Representation of the overfitting phenomenon

Regularization Techniques

Avoiding overfitting

L_2 regularization

Add a term to loss function, i.e,

$$\hat{\mathcal{L}}(\Omega, \theta) = \mathcal{L}(\Omega, \theta) + \lambda \,\text{complexity}(\theta), \tag{14}$$

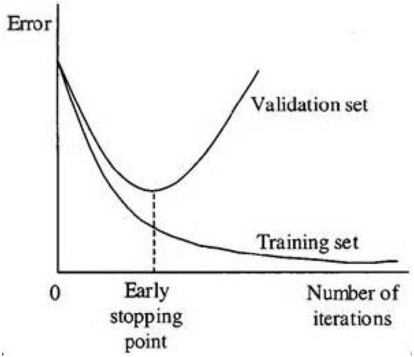
where λ is the regularization hyperparameter, and,

$$\text{complexity}(\theta) = \|\theta\|_2^2 = \sum_{w \in \theta} w^2. \tag{15}$$

Regularization Techniques

Avoiding overfitting II

- Early stopping



- Data augmentation

Testing concept

Save data to evaluate

- Evaluate model
  - Quantiative metrics
  - Qualitative
- Compare with state of the art

False evaluation metrics

Never train with `test` split!

Chapter Outline

3. Chapter 3. Referring Expression Comprehension
   - Problem Formulation
   - Training
   - Evaluation Techniques
   - Related Work

Referring Expression Comprehension

Reminder

Figure: Parent holding umbrella

- Input 1: Image
- Input 2: Referring Expression
- Output: Segmentation

Datasets

Subtitle

- RefCOCO
  - Generated from MS COCO dataset with a two-player game
  - 142,209 samples (in 19,994 images)
- RefCOCO+
  - Location information disallowed
  - Similar size
- RefCOCOg
  - Only non-trivial elements
  - 104,560 samples (in 26,711 images)
- CLEVR-REF+
  - Images generated synthetically (data augmentation)

Loss Functions

Cross entropy

The “difference” between predicted and ground truth pixels is computed as,

$$\mathrm{CE}(p, \hat{p}) = -(p \log \hat{p} + (1 - p) \log(1 - \hat{p})). \tag{16}$$

Intuitive interpretation

Taking into account that $p \in \{0, 1\}$, the loss function can be rewritten as follows,

$$\mathrm{CE}(p, \hat{p}) = \begin{cases} -\log(1 - \hat{p}) & p = 0 \\ -\log \hat{p} & p = 1. \end{cases} \tag{17}$$

Loss Functions

Cross entropy variants

Modifying cross entropy:

- Weighted Cross Entropy (WCE)

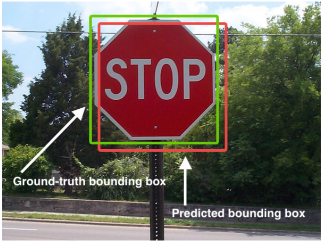
$$\mathrm{WCE}(p, \hat{p}) = -(\beta p \log \hat{p} + (1 - p) \log(1 - \hat{p})). \tag{18}$$

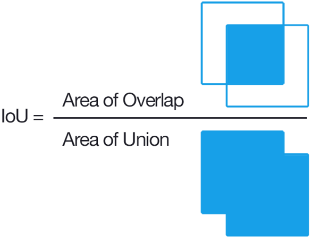
- Balanced Cross Entropy (BCE)

$$\mathrm{BCE}(p, \hat{p}) = -(\beta p \log \hat{p} + (1 - \beta)(1 - p) \log(1 - \hat{p})). \tag{19}$$

Loss Functions

Intersection over Union or Jaccard Index





Mathematical definition

Given the predicted segmentation A and the ground truth B, the Jaccard index is defined as follows,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{20}$$

Loss Functions

Dice Loss

Intersection over Union as loss function?

Cannot be used directly as a loss function since optimization will be infeasible (think about non-overlapping bounding boxes).

So, we define, Dice Loss (DL) as folows,

$$\mathrm{DL}(p, \hat{p}) = 1 - \frac{2 \sum p_{h,w} \hat{p}_{h,w}}{\sum p_{h,w} + \sum \hat{p}_{h,w}}. \tag{21}$$

Loss Functions

And much more

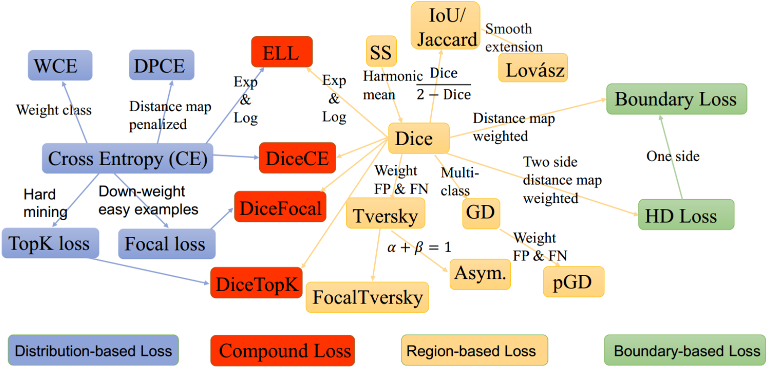


Figure: Overview of loss functions

Quantitative Measures

Derived from Intersection over Union

- Overall IoU, defined as,

$$\text{Overall Intersection over Union (IoU)} = \frac{\sum_{i=0}^{N} I_i}{\sum_{i=0}^{n} U_i}, \tag{22}$$

  where I_i and U_i correspond to the i-th intersection and union (respectively) between the prediction and the ground truth.

- Mean IoU, defined as,

$$\text{Mean IoU} = \frac{1}{N} \sum_{i=0}^{N} \text{IoU}_i. \tag{23}$$

- Precision at Threshold: judge as true/false positive using the IoU.

Qualitative Evaluation

- Analyze with your eyes
- Be smart
- Not numeric but useful

Multimodal Embedding

Joint space

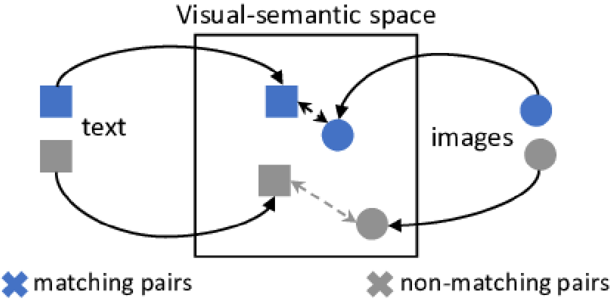


Figure: Multimodal embedding visual-semantic space

Modular Models

Models

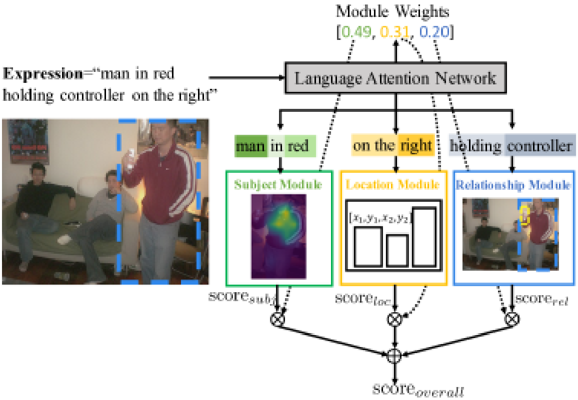


Figure: Modular Attention Network (MAttNet)

Graph Generation

Models

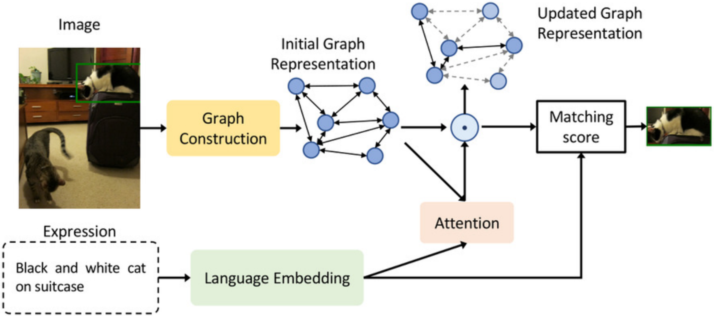


Figure: Summary representation of graph-based models

Chapter Outline

4 Chapter 4. Models
- Referring Expression Comprehension
- Speech Recognition

Base Architecture

Topology

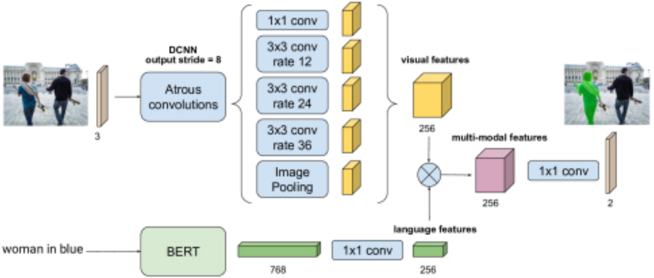


Figure: Referring Expression for Video Object Segmentation (RefVOS)

Base Architecture

Image Encoder

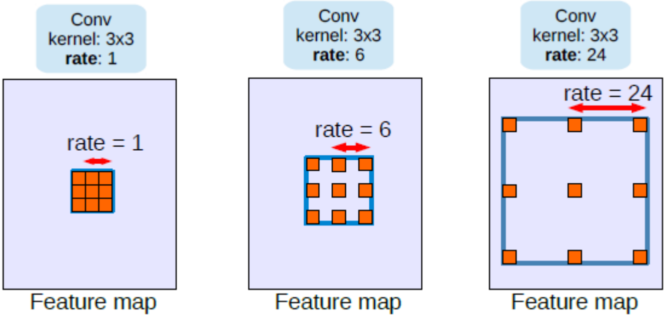


Figure: Atrous convolutions examples with filter size 3×3

Base Architecture

Language Encoder

- Bidirectional Encoder Representations from Transformers (BERT)
- Based on the Transformer model
- State of the art (even in computer vision!)

Base Architecture

Multimodal Embedding

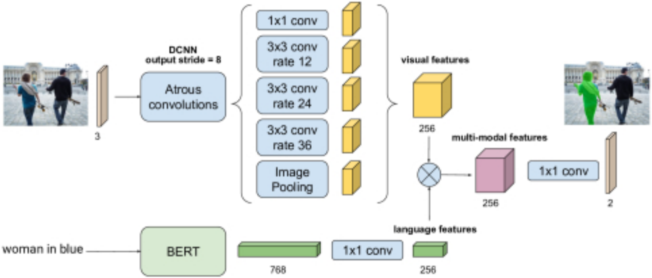


Figure: Referring Expression for Video Object Segmentation (RefVOS)

Model Iterations

Loss functions: cross entropy variants

- Weighted Cross Entropy
- Balanced Cross Entropy

Useless learning

- Similar functions (loss objective)
- Small (near 0) partial derivatives
- Very little learning

Model Iterations

Loss functions: Dice Loss I

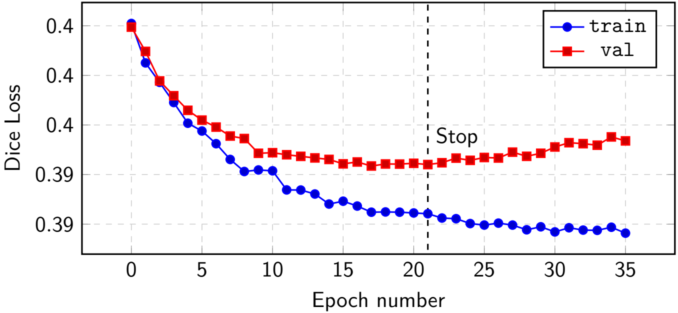


Figure: Training graph with Dice Loss

Model Iterations

Loss functions: Dice Loss II

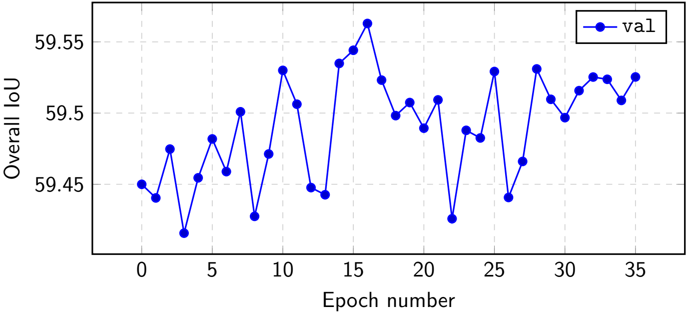


Figure: Overall IoU graph with Dice Loss

Model Iterations

Loss functions: Tversky Index

- Similar results
- Non significant improvements

Model iterations

Fusion strategies performance in RefCOCO dataset

	RefCOCO		
Strategy	`val`	`testA`	`testB`
Addition	56.60	60.87	51.29
Multiplication	**59.45**	**63.19**	**54.17**
Concatenation	55.12	58.88	49.59
Projection	Infeasible	Infeasible	Infeasible
Projection v2	21.08	-	-

Model Iterations

Multimodal embedding: Projection

It is possible to define applications to map features to a vector space of common dimension J. That is, the application ϕ is defined,

$$\begin{aligned} \phi\colon \mathbb{R}^D \times \mathbb{R}^{D\times J} &\longrightarrow \mathbb{R}^J \\ (V, W_v) &\longmapsto \phi(V, W_v) := W_v V, \end{aligned} \tag{24}$$

which maps the visual features V to the joint space $\mathbb{R}^J$ via W_v. In the same way, the application ψ is defined,

$$\begin{aligned} \psi\colon \mathbb{R}^d \times \mathbb{R}^{J\times d} &\longrightarrow \mathbb{R}^J \\ (L, W_l) &\longmapsto \psi(L, W_l) := W_l L, \end{aligned} \tag{25}$$

which maps the language features L to the joint space $\mathbb{R}^J$ via W_l.

Model Iterations

Multimodal embedding: Projection v2

Computationally infeasible

$W_v \in \mathbb{R}^{D \times J}$, and $D \times J$ is really, really big (order of billions).

Solution:

- Use same parameters for each slice in the depth of the visual features.
- For each slice V^i, use same weigth matrix W_v and embed,

$$\tilde{V}^i = W_v V^i. \tag{26}$$

Model Iterations

Multimodal embedding: Projection v2 II

This underperforms previous model (21.08 overall IoU). Caused by:

- Loss of spatial information
- Meaningless transformation to visual information
- Adding unnecessary non-pretrained parameters

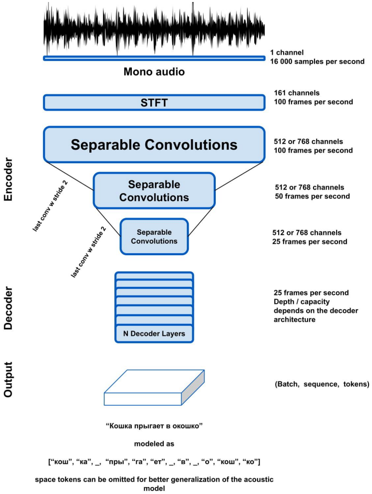
Training process

Different techniques

Several optimization methods exists, and can be changed, but this won't change the ability of the model to learn necessarily.

Speech to Text

Silero Model



Chapter Outline

5 Chapter 5. Results and Comparison
- Quantitative Evaluation
- Qualitative Evaluation

Overall Intersection over Union

Model comparison

		RefCOCO			RefCOCO+		
Method	**Paper**	val	testA	testB	val	testA	testB
ASGN	[Qiu+20]	50.46	51.20	49.27	38.41	39.79	35.97
BRINet	[Hu+20]	61.35	63.37	59.57	48.57	52.87	42.13
CAC	[Che+19b]	58.90	61.77	53.81	-	-	-
CMPC	[Hua+20]	**61.36**	**64.53**	**59.64**	**49.56**	**53.44**	**43.23**
CMSA	[Ye+21]	58.32	60.61	55.09	43.76	47.60	37.89
DMN	[Mar+18]	49.78	54.83	45.13	38.88	44.22	32.29
MAttNet	[Yu+18]	56.51	62.37	51.70	46.67	52.39	40.08
RefVOS	[Bel+20]	59.45	63.19	54.17	44.71	49.73	36.17
RMI	[Liu+17]	45.18	45.69	45.57	29.86	30.48	29.50
RRN	[Li+18]	55.33	57.26	53.95	39.75	42.15	36.11
STEP	[Che+19a]	60.04	63.46	58.97	48.18	52.33	40.41

Accuracy or Precision at 0.5

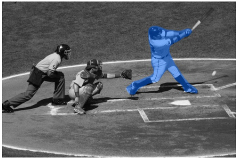
Model comparison

Method	Paper	RefCOCO			RefCOCO+		
		val	testA	testB	val	testA	testB
BRINet	[Hu+20]	71.83	75.09	68.38	-	-	-
CAC	[Che+19b]	77.08	80.34	70.62	-	-	-
CMAttErase	[Liu+19b]	**78.35**	**83.14**	**71.32**	68.09	73.65	58.03
CMPC	[Hua+20]	71.27	-	-	-	-	-
CMSA	[Ye+21]	69.24	73.87	64.55	45.48	51.41	37.57
FAOA	[Yan+19]	71.15	74.88	66.32	56.88	61.89	49.46
LGRAN	[Wan+19]	-	76.6	66.4	-	64.00	53.40
MAttNet	[Yu+18]	76.65	81.14	69.99	65.33	71.62	56.02
MMI	[Mao+16]	-	64.90	54.51	-	54.03	42.81
NMTree	[Liu+19a]	74.71	79.71	68.93	65.06	70.24	56.15
RefVOS	[Bel+20]	67.34	70.47	65.02	57.28	60.31	46.37
RMI	[Liu+17]	42.99	42.99	44.99	20.52	21.22	20.78
RRN	[Li+18]	61.66	64.13	59.35	37.32	40.80	32.42
STEP	[Che+19a]	70.15	-	-	-	-	-
ViLBERT	[Lu+19]	-	-	-	**72.34**	**78.52**	**62.61**

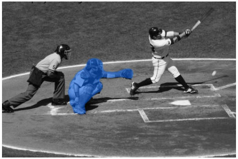
Study of Succesful Samples

Examples

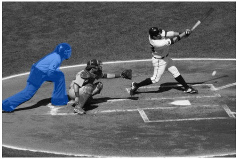
(a) Player with baseball bat



(b) Middle player with glove



(c) Man in the left



(d) Donuts with topping

(e) White background donuts

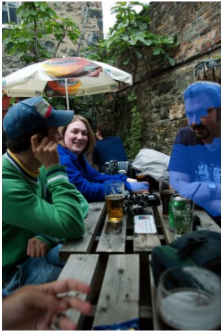
(f) White donut left behind

Study of Succesful Samples

Examples II

(a) Person in blue

(b) Person with watch



(c) Woman

Study of Failed Samples

Examples

(a) Left tennis racket

(b) Blond boy looking back

(c) Banana

(d) Statue

(e) Statue of a bird

(a) Woman holding hair dryer

(b) Hair dryer

(c) Tennis match referee

(d) Tennis match referee sitting behind

Chapter Outline

6 Chapter 6. Visualization
- User Interface
- Back End

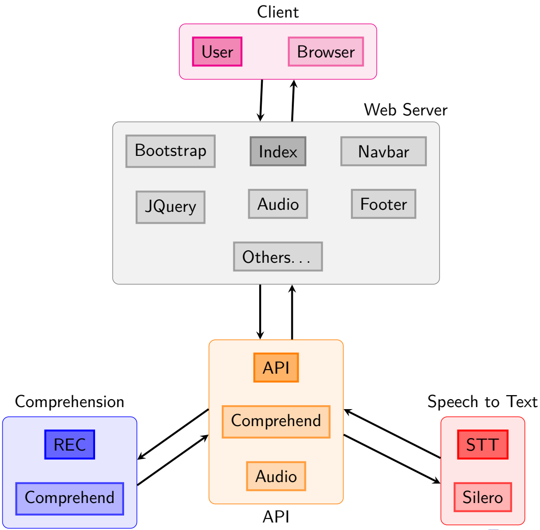
Website overview

Presentation of User Interface

- Responsive Web Design
- Accesibility
- Usage Example: `https://recomprehension.com`

Back end

Graphic representation



Chapter Outline

7 Chapter 7. Project Analysis
- Planning and Scheduling
- Cost Analysis (view thesis)
- Environmental Impact (view thesis)

Table of Activities

Main activities broken down into tasks

Code	Activity	Start	End
A	**Learn basics of ML/DL**	Oct.	Jan.
A1	ML course [Ng20]	-	-
A2	DL lectures from UPC [Gir20]	-	-
A3	Stanford CS231n: CNNs for Visual Recognition [LKX20]	-	-
A4	DL specialization [NKM20]	-	-.
B	**Learn thesis topic**	Dec.	Feb.
B1	Multimodal learning lectures [Gir20]	-	-
B2	Publications	-	-
B3	State-of-the-art papers on REC	-	-

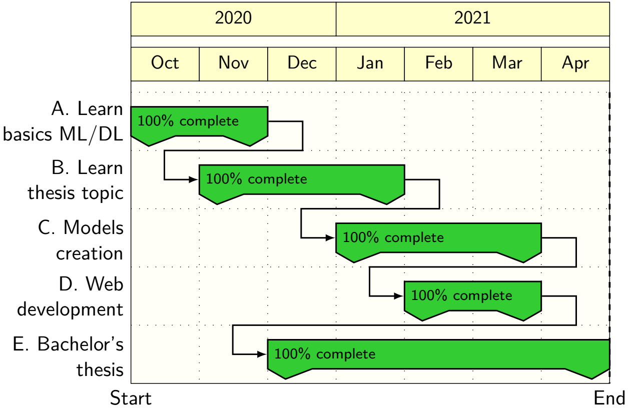
Table of Activities

Main activities broken down into tasks

Code	Activity	Start	End
C	**Models creation**	Jan.	Apr.
C1	Server usage	-	-
C2	Multiple iterations	-	-
C3	Generate test values	-	-
D	**Web development**	Feb.	Apr.
D1	Front end (HTML, CSS, JS)	-	-
D2	API creation (PHP)	-	-
D3	Web server configuration	-	-
D4	Publish website (domain, server)	-	-
E	**Bachelor's thesis**	Dec.	May
E1	Write thesis (LaTeX)	-	-
E2	Create presentation slides (LaTeX)	-	-
E4	Prepare presentation	-	-

Gantt Chart

Main activities



Chapter Outline

8. Chapter 8. Conclusions
   - Future work

Conclusions

General

- Introduction to the fields of
    - Machine Learning
    - Computer Vision
    - Natural Language Procesing
- Improve programming skills (Python, Pytorch)
- Web development (front end, back end API)

Future possibilities

For thesis and more

- Fix "black box" problem (reasoning process cannot be visualized)
  Ideal multi-step with **woman in red dress sitting on the right**:
  1. Find all the women present in the image, these objects will be the only solution candidates.
  2. From these women choose all those who wear a red dress.
  3. From this group select those that are seated.
  4. Finally, if there is more than one possibility, select the one on the right.
- Lack of high quality datasets
- Adapt to video (with temporal coherence)
- Leaderboard creation and objective evaluation of state of the art

Thanks

Thank you!