Recurrent Instance Segmentation with Linguistic Referring Expressions

Alba María Herrera Palacio

Abstract

The goal of this work is segmenting the object in an image or video which is referred to by a linguistic description (referring expression). We propose a deep neural network with recurrent layers that output a sequence of binary masks, one for each referring expression provided by the user. The recurrent layers in the architecture allow the model to condition each predicted mask on the previous ones, from a spatial perspective within the same image. Our multimodal approach uses off-the-shelf architectures to encode both the image and the referring expressions. The visual branch provides a tensor of pixel embeddings that are concatenated with the phrase embeddings produced by a language encoder. We focus our study on comparing different configurations to encode and combine the visual and linguistic representations. Our experiments on the RefCOCO dataset for still images indicate how the proposed architecture successfully exploits the referring expressions to solve a pixel-wise task of instance segmentation.

Index Terms

Instance segmentation, object grounding, recurrent neural networks, linguistics

I. Introduction

INSTANCE segmentation with natural language expressions is a challenging problem with implications in the fields of computer vision and natural language processing. The goal is to segment the referent, i.e., the target object referred to by a referring expression, in an image. The referring expressions used for this task can take any form of linguistic description. They can contain appearance attributes (e.g. *"red"*), actions (e.g. *"singing"*), relative relationships (e.g. *"left"*), etc. and are not limited to object categories, as it is important to be able to distinguish between instances of objects without ambiguities. Instance segmentation with referring expressions can be understood as an extension of semantic instance segmentation, where a binary mask and a categorical label are assigned to each object in an image (see comparison in Figure 1). Since a natural way of human interaction is through referring expressions, the ability to comprehend those is necessary in human-computer real-world interaction scenarios.

Existing works [1]–[3] in this area separately represent the linguistic expression and the input image, typically using recurrent neural networks (RNN) and convolutional neural networks (CNN), respectively. Afterwards, in order to obtain a pixel-wise segmentation mask, both representations are concatenated and further processed.

The goal of this project is to develop a multimodal neural network architecture to recurrently segment target objects by linguistic referring expressions. In other words, given an image and a referring expression for each of the instances to be segmented, we aim to obtain pixel-level masks of the referents. The proposed architecture consists of: (i) a vision encoder, which extracts visual features of a frame, (ii) a language encoder, which adds linguistic information to the model by using a pre-trained natural language processing model to extract language features for the referring expressions (phrases), and (iii) a

Author: Alba María Herrera Palacio, albaherrerapalacio@gmail.com
Advisor 1: Xavier Giró-i-Nieto, Universitat Politecnica de Catalunya
Advisor 2: Carles Ventura, Universitat Oberta de Catalunya
Advisor 3: Carina Silberer, Universitat Pompeu Fabra
Thesis dissertation submitted: September 2019

(a) Input image (b) Of class *"person"* (c) Of class *"person"* (d) Referring expression *"left woman in blue"*

Figure 1: Comparison between semantic the tasks of: (b) object segmentation, (c) object instance segmentation and (d) segmentation from natural language expressions.

Figure 2: Example of the semi-supervised video object segmentation problem using linguistic referring expressions from [7]

binary mask vision decoder, which uses the image and phrase embeddings from the vision and language encoders, respectively, to generate the pixel-level masks of the target objects. The visual encoder and decoder are inspired by RVOS, an existing Recurrent network for multiple object Video Object Segmentation [4]. To obtain embeddings for the referring expressions we use the language encoder BERT [5]. We use RefCOCO [6] dataset by University of North Carolina (UNC) which provides images with pixel-level segmentation masks along with multiple referring expressions for each referent to train and evaluate the model.

This master thesis has been developed in the framework of a wider and more ambitious project of video object grounding with referring expressions, as depicted in Figure 2. For this reason, in addition to the core contributions over still images, this thesis also includes an exploratory study about the potential of RVOS for solving the final task. As in [7], we used the off-the-shelf tool of MAttNet [8] to predict a binary mask over the first frame of the video sequence. Then, we used RVOS to propagate this initial mask through the rest of the video sequence. The result of this study was peer reviewed and accepted for presentation as poster and publication in the ACM Multimedia 2019 Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA) [9].

II. STATE-OF-THE-ART

This master thesis has been developed in the intersection between language and video object segmentation, mostly addressing the basic case of instance segmentation with referring expressions in still images. This section is structured in three parts that address these three different fields.

A. Natural language processing

1) Word Embeddings: Natural Language Processing (NLP) models usually require inputs in the form of numerical vectors. In the past, words have been represented as one-hot vectors which represent the numerical value of the word in the vocabulary and grow with the vocabulary size, words are thus represented in a discrete way and cannot be compared mathematically to estimate their similarity, e.g. *"king"* is more similar to *"queen"* than to *"computer"*. Nowadays, fixed-length word embeddings are used, which have the ability to generalize due to semantically similar words having similar vectors, and which are computationally more efficient.

From a linguistic perspective, word embeddings are dense vector representations of words in a high dimensional space. Those embeddings are used as geometric representations of the meaning of words, and hence one can compare words with each other my means of mathematical operations.

This approach was successfully implemented with deep neural networks such as the word2vec model [10]. Afterwards, more advanced models appeared, which not only captured a static semantic meaning, but also a contextualized meaning. This allowed the embedding of a word to have a different vector representation based on the context. This type of word embeddings are more suitable for most of NLP tasks.

2) Sequential Models: Usually, sequence-to-sequence tasks in which a sequence is mapped to another sequence, such as machine translation, are performed using an encoder-decoder model, as illustrated in Figure 3. The encoder takes the sequence of input tokens[1], converts it to some intermediate representation, then passes that representation to the decoder which produces the output sequence. These models are trained to maximize the likelihood of generating the correct output sequence.

Conventional sequential models seemed to be well suited for encoding and decoding language and were widely used due to the sequential nature of language. Nevertheless, there are a few shortcomings of conventional sequential models, such as creating variable length long-range dependencies that need to be carried along while reading the source and generating the target, see Figure 4, since information is available at different time steps between the encoder and decoder.

[1]From a linguistic perspective, a token is a string of contiguous characters between two spaces or between a space and punctuation marks. It can also be a real number or a number with a colon (e.g. *"10:30"*).

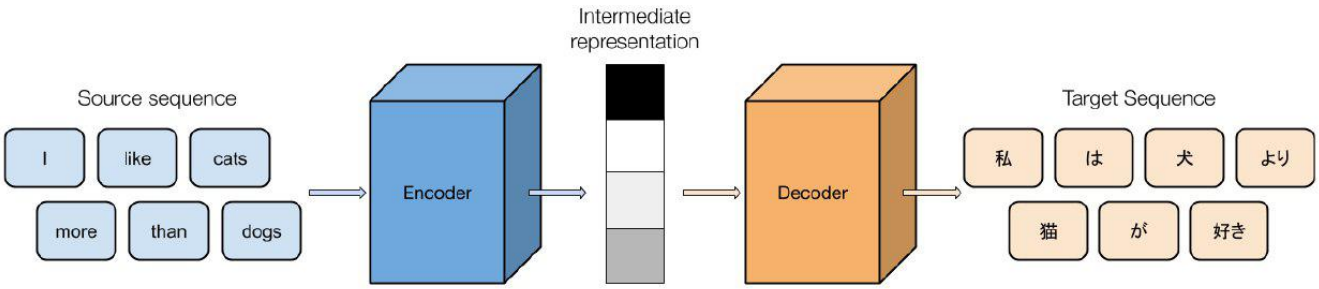


Figure 3: Machine translation example for an encoder-decoder architecture.

3) Attention-based Models: The Transformer [11] got rid of the recurrent component and introduced an encoder-decoder architecture model which only uses attention mechanisms to focus on relevant information based on what it is currently processing. Both, self-attention (encoder on encoder) and attention on the decoder are used, which allow the decoder to access at once the entire sequence information, rather than sequentially.

The attention weights are the relevance of the encoder hidden states (values) in processing the decoder state (query) and are calculated based on the encoder's hidden states (keys) and the decoder's hidden state (query). The decoder selectively extracts the information it needs during decoding, in other words, chooses among the encoder's hidden states by weighting them. Eliminating the sequential dependency on previous tokens increases the effectiveness in modeling long-term dependencies among tokens in a temporal sequence. As illustrated in Figure 5, the dependency that the Transformer has to learn is independent of the length and order of the source and target sentences.

Since the meaning of a word is context-dependent, e.g. two identical words may have different meanings depending on the context (polysemy), the vector which represents any word should also change depending on the context. The first attempt to take the surrounding context into account was ELMo, Embeddings from Language Models [12], where the hidden states of an LSTM for each token are used to compute a vector representation of the meaning of each word. But, since there is a single LSTM for the forward language model and backward language model each, neither LSTM takes both the previous and subsequent tokens into account at the same time.

For some words, their meaning might only become apparent when you look at both the left and right context simultaneously. BERT, a Bidirectional Encoder Representations from Transformer [5], accounts for this problem by producing word embeddings taking both the previous and next tokens into account.

BERT uses the Transformer architecture to incorporate information from the entire input sentence. But in contrast to LSTMs, this architecture does not naturally take the order of the tokens into account, so BERT uses positional embeddings to encode the location of each word within the sentence. Although BERT can also take as input single-sentences, it is trained on and expects concatenated sentence pairs, segmentation embeddings specify which sentence each token belongs to. So, for a given token, its input embeddings are the sum of the token embeddings, the segmentation embeddings, and the position embeddings

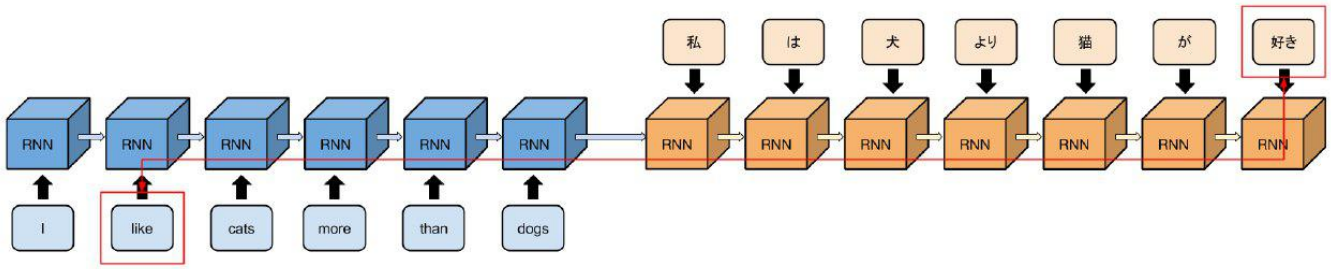


Figure 4: Machine translation example for a recurrent neural network.

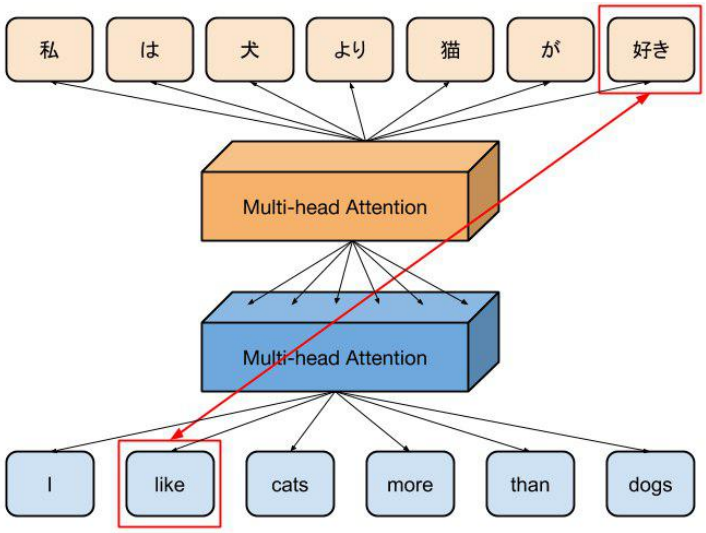


Figure 5: Machine translation example for a transformer.

(see Figure 6).

These models can be fine-tuned on a specific task or used to extract contextualized word embedding to be used as high quality linguistic features.

In this thesis, we will use BERT embeddings to encode the linguistic references to the object instances.

B. Visual instance segmentation from natural language expressions

The goal of visual grounding with referring expressions is to localize specific objects in an image by means of a natural language description of each object. Most existing approaches rely on scored bounding box proposals to determine the correct localization region with a bounding box. To obtain a more precise result, instance segmentation was proposed, which produces segmentation masks for objects described by a natural language expression instead of bounding boxes.

This problem was firstly introduced in [1], where a CNN and an LSTM are used to extract visual and linguistic features, respectively. The fusion of the visual and linguistic representations was done by concatenating the LSTM output to the visual feature map at each spatial location. Then the low resolution output is up-sampled using deconvolution layers to yield the pixel-wise segmentation mask.

Instead of segmenting the image based only on a phrase embedding (i.e., referring expression or parts thereof), [13] exploits word-to-image interactions, directly combining visual features with each word feature, obtained from a language LSTM, to recurrently refine the segmentation results. [14] employs in a sequential manner both, a concatenation strategy to merge linguistic and visual features, and the computation of a dynamic filter, whose response is directly related to the presence of the object at a given spatial coordinate in the image. In [2], a convolutional LSTM (ConvLSTM) [15] model encodes and fuses visual and linguistic representations and outputs a rough localization of the referent. A recurrent refinement module then uses the fused representation and pyramidal image features to refine the segmentation by adaptively selecting and fusing image features at different scales.

To adaptively focus on informative words in the referring expression and important regions in the input image, cross-modal self-attention (CMSA) module [16] captures the long-range dependencies between linguistic and visual contexts to produce

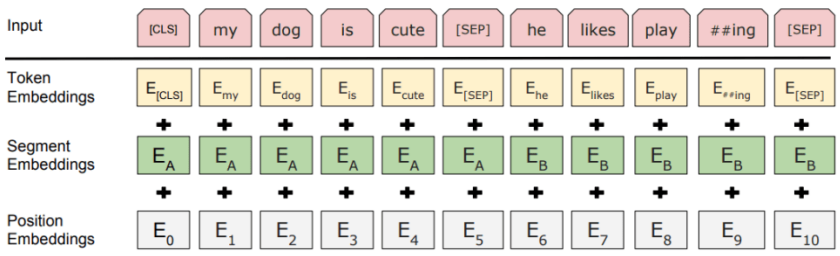


Figure 6: BERT input representation from [5].

(a) Input image (b) *"white horse left"* (c) *"first horse right"*

Figure 7: Example of a segmentation from unambiguous natural language expressions for instances of the class *"horse"*.

multimodal features. Moreover, a gated multi-level fusion module selectively integrates multi-level self-attentive features, corresponding to different levels in the image, which effectively capture fine details for precise segmentation masks.

In MAttNet [8], a language attention network decomposes referring expressions into three components, one for each visual module (subject, location, and relationship modules), and maps each of them to single phrase embeddings. Given candidate objects by an off-the-shelf instance segmentation model and a referring expression, the visual module dynamically weights scores from all three modules to output overall scores. Two types of attention are used: (i) language-based attention, which learns the module weights as well as the word or phrase attention that each module should focus on, and (ii) visual attention, which allows the subject and relationship modules to focus on relevant image components.

C. Semi- and weakly-supervised video object segmentation

Semi-supervised video object segmentation methods [4], [17] use a binary mask of the target object manually annotated in the first frame which is propagated to successive frames. This set up is also referred as one-shot video object segmentation, as one example mask of the object to segment is provided. However, the manual annotation of a pixel segmentation for the first frame is tedious and time-consuming. Another approach, referred to as interactive segmentation, is the segmentation of target objects whose locations are roughly indicated by human interaction. Therefore, costly pixel-level masks are replaced by point clicks [18] or scribbles [19] to specify the target object.

Our scope is to use an even simpler form of interaction, as it is based on natural language. Similarly to the still images task, a linguistic referring expression is the only form of supervision. In this case, the expression can also refer to the motion of the object, given the video nature of the problem. Up to the authors knowledge, the task of video object segmentation with referring expression was firstly addressed in by Khoreva et al. [7], who collected linguistic descriptions of target objects over the DAVIS 2016 and 2017 [20] datasets for video object segmentation. Given a referring expression and each frame in the video sequence, the authors in [7] used the MAttNet [8] model for still images to generate target object bounding box proposals for each frame. To mitigate temporally inconsistent and jittery box predictions, they enforced temporal consistency, such that bounding boxes are coherent across frames. Finally, they used the obtained box predictions of the target object to recover detailed object masks in each frame applying a CNN-based pixel-wise segmentation model. We used a similar approach in our exploratory study on video object segmentation presented in Section V, but instead of using the bounding box proposals provided by MAttNet in all frames, we only use the mask that MAttNet provides for the first frame. This first mask is propagated to the posterior frames as in the classic semi-supervised video object segmentation works.

III. Method

We propose a multimodal model to recurrently segment target objects identified by linguistic referring expressions. The proposed deep neural network is depicted in Figure 8. The architecture is an extension RVOS-S, the model that deals with single frames in the RVOS video object segmentation model presented in [4]. The input of the network consists of: (i) a referring expression for each of the instances to segment (referents), and (ii) an image in the RGB color-space. The proposed model consists of two encoders (vision and language) whose pixel and phrase embeddings are concatenated and fed to a binary mask visual decoder.The output is a pixel-level mask for each referent. Figure 7 shows an example of the linguistic and visual inputs for a referent, and an expected output.

A. Referring expression encoder

BERT [5] is used with a feature-based approach, without fine-tuning any of the parameters, to obtain representations for the referring expressions, i.e, inputs to our language encoder, as seen at the top branch of Figure 8. By extracting the activations from one or more layers we obtain a sequence of contextualized token embeddings (hidden states) and a pooled output (phrase embedding), as illustrated in Figure 9. In the following subsections it is detailed how contextualized phrase embeddings, inputs for the language encoder, are obtained from them.

The implementation of BERT embeddings used in our work[2] actually offers different off-the-shelf model variations. The

[2]https://github.com/huggingface/pytorch-transformers

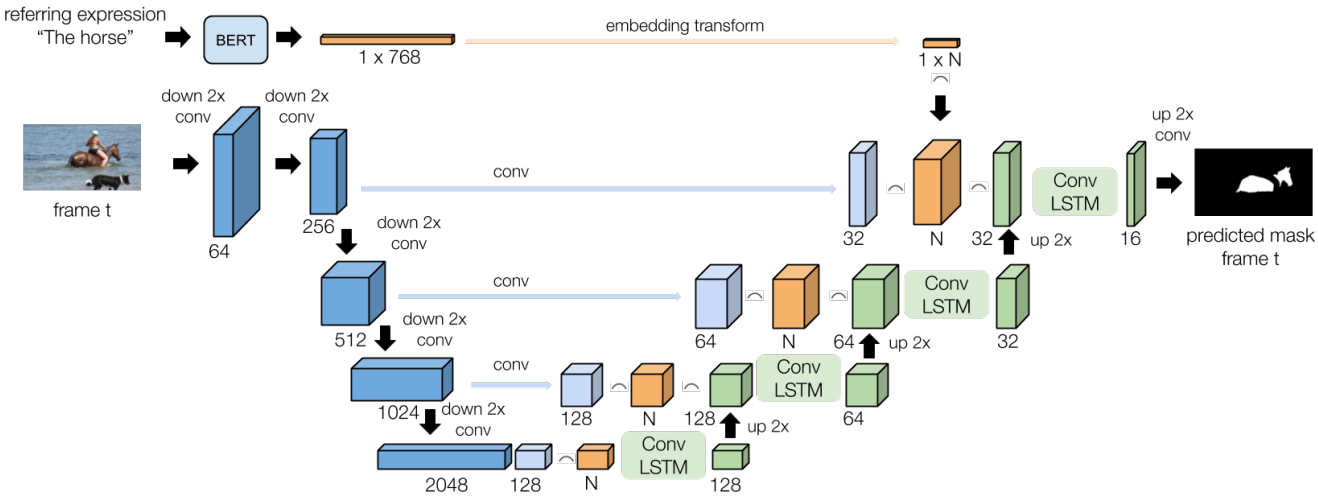


Figure 8: Our proposed recurrent architecture for recurrent instance segmentation with linguistic referring expressions. The figure illustrates a single forward pass, predicting only the mask of one instance for an image.

ones available to extract embeddings from English text are:

- **Base model:** 12 encoder layers (transformer blocks), 768 hidden units and 12 attention heads.
- **Large model:** 24 encoder layers (transformer blocks), 1024 hidden units and 16 attention heads.

We use the `bert-base-cased model`, which corresponds to the base model and ignores casing.

A tokenizer is provided to pre-process the raw text data, since BERT is a pre-trained model that expects input data in a specific format. The tokenization steps for sentences are:

- **Text normalization:** Convert all whitespace characters to spaces. Additionally, lowercase the input and strip off accent markers for the uncased model.
- **Punctuation splitting:** Split all punctuation characters[3] by adding a space around them.
- **WordPiece tokenization:** Apply space tokenization to the output of the above procedure. Then tokenize each word separately. The tokenizer first checks if the whole word is in the vocabulary. If it is not, it tries to break the word into

[3]Punctuation characters are defined as any ASCII character different from a letter, number o space, or with a P* Unicode class.

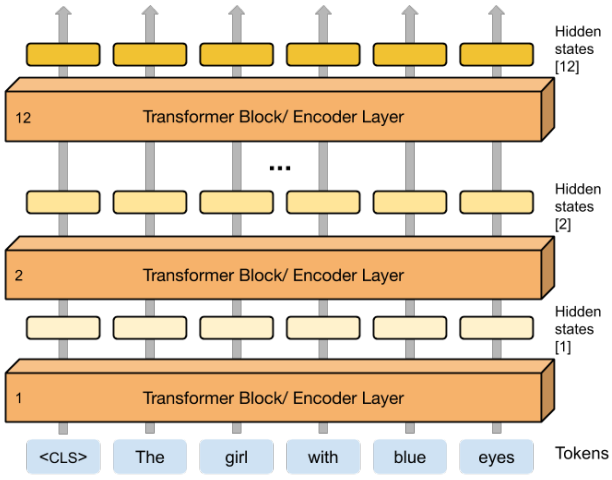


Figure 9: General scheme of BERT outputs used to generate contextualized embeddings. The output of each layer corresponding to a token can be used as a feature representing that token.

the largest possible subword tokens contained in the vocabulary, that will retain some of the contextual meaning of the original word. As a last resort it will decompose the word into individual characters. Note that because of this, we can always represent a word as, at the very least, the collection of its individual characters.

1) Pooled output: The pooled output encodes a whole sequence as a single vector. Every sequence always starts with the special classification token *[CLS]*, see Figure 9, which stands for classification. The last the state of the last hidden layer that encodes the *[CLS]* token is used as the aggregate sequence representation for classification tasks. The pooled output corresponds to that hidden state that is trained on the task of predicting the next sentence. As BERT authors remark, this output is usually not a good summary of the semantic content of the input.

2) Encoded layers: At the output of the last layer of the model, a set of contextualized embeddings (hidden states for this model) is stored in an four-dimensional tensor that has four dimensions, in the following order: layer number (12 layers/hidden states), batch number (1 sentence), word/token number in each sentence, and hidden unit/feature number (768 features). There are multiple strategies to obtain phrase contextual embeddings from them, which involve: (i) word embedding combination strategy, such as averaging, concatenating, pooling, etc., and (ii) sequence of hidden-states/layers used, such as last four, all, last layer, etc. In our work, we average the last hidden layer of each token producing a single 768 length vector for each referring expression.

3) Dimensionality reduction: A single embedding for each of our referring expressions is provided by the language encoder. To avoid memory problems while training the model and balance the dimension of the language and visual embeddings, we tried reducing the dimensionality of the textual embeddings. In our experiments we explore two ways of dimensionality reduction: principal component analysis (PCA) [21], and a learned linear projection, adding a dense layer that is trained with the model.

B. Image encoder

The still image is fed to the network as an RGB color image and encoded with ResNet-101 [22] model pre-trained on ImageNet [23]. The ResNet architecture is truncated at the last convolutional layer, thus removing the last two layers (pooling layer and classification layer). This architecture is finetuned for the specific task being solved, i.e. object segmentation by using referring expressions. The output of each convolutional block is used as an image feature, which provides a set of visual features at different resolutions, as shown in dark blue at the left of Figure 8. This visual encoding scheme was adopted from RVOS-S [4], which at the same time was inspired by the former RSIS [24] model for semantic instance segmentation over still images.

C. Mask decoder

We explore several strategies to combine the language embeddings with multi-resolution visual features in the decoder. We wanted to preserve both linguistic and visual information in the segmentation, while avoiding high memory requirements during the training of the model. Therefore, to keep the inherent spatial information in the visual features when segmenting an instance, for each resolution, we concatenate the corresponding language embedding to each feature map along the channels' dimensions (depth) of the visual tensors. This allows every pixel embedding to receive the whole representation of the language information.

The recurrent architecture of the decoder, inherited from RVOS-S, allows to condition the predicted masks with those masks predicted from previous referring expressions over the same image.

The ConvLSTM [15] layers used in the decoder contain a state memory that has the potential to remember previous predictions. This spatial recurrence, which is explored in our work, could also facilitate the future extension of the work to video. The original RVOS [4] treated with the same ConvLSTM [15] layers both spatial and temporal recurrences. This allows the model to condition each predicted mask on the previous ones, whether from a spatial perspective within the same image, from a temporal view, when the state of the recurrent layer is propagated through the frames of a video, or from a spatio-temporal view, where both spatial and temporal recurrence are used. This possible extension to video, while not covered in this master thesis, was the main reason to adopt the RVOS architecture as a starting point.

D. Loss

The assignment between predicted masks and ground truth is based on the order in which the referring expressions for each referent are processed by the model. Similarly to [4], [24], the cost function is defined as the soft Intersection over Union score between the predicted mask m_1 and the ground truth mask m_2 for a given referent.

$$\mathrm{sIoU}(m_1, m_2) = 1 - \frac{\sum_{i=1}^{M} m_{1,i} m_{2,i}}{\sum_{i=1}^{M} m_{1,i} + m_{2,i} - m_{1,i} m_{2,i}} \tag{1}$$

IV. EXPERIMENTS

This section presents the results obtained with the proposed architecture for instance segmentation with linguistic referring expressions. The experiments show how the introduction of the referring expression encoder successfully conditions the mask to predict. Our study also includes several ablation studies and a comparison with the state of the art.

A. Experimental Setup

1) Dataset: Our experiments use RefCOCO dataset [6], which provides both pixel-level masks and linguistic referring expressions for an image in complex real world scenes. The referring expressions for objects were collected in an interactive way with the ReferItGame proposed by [25]. Since each image contains multiple objects of the same category, the referrer must provide unambiguous relevant referring expressions, see an example in Figure 7.

RefCOCO contains $142,210$ referring expressions for $50,000$ referents (from 80 MSCOCO categories). These referents are depicted in $19,994$ images from the MSCOCO [26] dataset and the segmentation masks of the referents. We adopt the training, validation, and testA and testB splits provided by the University of North Carolina (UNC), which have respectively $42,404$, $3,811$, $1,975$ and $1,810$ referents. There is no overlap between train, validation and test images. Since half of the referents are people, people-vs-objects splits are used for testing, where testA images contain multiple people and testB multiple instances of all other objects.

2) Training details: The RefCOCO dataset provides several referring expressions for each segmented object. However, we only use one referring expression per referent in each epoch because we observed that using multiple ones was harmful for the performance of our model. This behaviour can be explained because of the spatial recurrence (memory) that RVOS-S contains. RVOS-S was actually designed for instance segmentation, in such a way that each pixel in the image can only belong to a single instance. This would be contradictory with using multiple referring expressions for the same referent.

Given a referent, referring expressions are chosen in the following way:

- During training, a referring expression is chosen randomly in each epoch. The training data consequently changes every epoch and acts as a sort of regularization.
- For validation and testing, we always use the first referring expression provided by the dataset, which speeds up validation and maintains consistency in the results. Even though not all referring expressions are used, embeddings obtained from referring expressions of the same referent are similar enough to obtain results representative of the whole dataset to compare with state-of-the-art methods. This hypothesis is validated comparing the results from testing with the first referring expression to the ones using a random one, which shows a minimal variation (on average less than 1% in overall IoU).

Given an input image, we resize it to $240x427$ pixels and keep the maximum number of referents to 6. The network is trained with ADAM optimizers with a weight decay of 10^{-7} and an initial learning rate of 10^{-4} (language encoder and decoder) and 10^{-7} (image encoder). We do not fine-tune the pre-trained BERT model during training. However, ResNet-101, which is used by the visual encoder, is fine-tuned with the architecture.

Experiments have been implemented with Pytorch in Python3.

3) Metrics: The quality of the predicted binary masks is evaluated with the overall and instance-level intersection-over-union (IoU) and $precision@X$. Overall IoU, referred as micro-average in NLP, calculates the total intersection area between predicted segmentation masks and ground truth divided by total union area accumulative over all the referents. A well-known issue regarding this IoU measure is its bias toward object instances that cover a large image area. To evaluate how well the individual instances are segmented independently of their area, we evaluate the segmentation on an instance-level, i.e., the global IoU is obtained averaging across all referents, where for each referent IoU is calculated as the ratio between intersection and union of the prediction and the ground truth. The latter IoU measure, also known as macro-average in NLP, is the same used by the model to calculate the loss. The percentage of referents with an IoU score higher than a given threshold X is the $prec@X$ metric measure, set in the experiments to $X \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

B. BERT embedding and dimensionality reduction

We tested the impact of different ways to encode the referring expressions by the BERT-based language encoder, described in Section III-A. The results on the RefCOCO dataset with a batch size of 32 images, randomly sorting the referents, are given in Table I showing the following comparisons: (i) BERT embedding: pooled output or encoded layers, (ii) transform for dimensionality reduction: None, using PCA or a trained linear projection, with a dimension of 768 or 128 or 64.

Even though both linguistic representations obtain similar results, encoded layers is slightly better. In general, results improve when applying a dimensionality reduction strategy. Overall, an embedding of dimension 64 works better. The best results are achieved with PCA. Probably the reason why the linear layer approach does not achieve better results is that it has too many parameters to learn. To see if the results could be improved, the linear layer can be initialized with the PCA values.

Instance IoU, which gives the same importance to each referent independently of their size, obtains higher results across all the configurations. Overall IoU, which does not take the size of the ground truth or the segmentation into account, obtains worse results since it is greatly affected by cases where the output mask selects large areas of the image.

Table I: Results in RefCOCO for different embeddings configurations.

BERT embedding	**Transform**	**Embedding dimension**	Instance IoU			Overall IoU		
			val	**testA**	**testB**	**val**	**testA**	**testB**
Pooled output	None	768	25.57	30.30	21.92	23.23	26.85	20.38
	PCA	128	27.70	32.17	24.27	25.00	28.25	22.82
		64	29.59	33.47	26.66	33.65	39.02	30.17
	Linear	128	26.44	30.88	23.50	23.76	27.38	21.67
		64	26.43	31.00	22.43	24.25	27.67	21.02
Encoded layers	None	768	26.86	31.32	23.04	24.51	27.85	21.58
	PCA	128	23.24	27.06	20.13	20.99	24.22	18.89
		64	**39.79**	**45.31**	**34.04**	**35.70**	**40.28**	**31.28**
	Linear	128	27.57	31.85	23.74	24.95	28.25	22.19
		64	36.36	42.27	31.94	32.23	37.06	28.60

C. Order of the referents and batch size

We perform additional experiments to further investigate the effect of different configurations to train the model. Table II shows the impact of varying the batch size and the way referents are fed to the model. The latter can be sorted: (i) by area, (ii) randomly, each time an image is forwarded its objects are fed in a different order, which acts as a sort of data augmentation, and (iii) as in the RefCOCO dataset.

Table II: Results in RefCOCO with BERT encoded layers reduced with PCA to dimension 64 as linguistic embeddings.

Referent order	**Batch size**	Instance IoU			Overall IoU		
		val	**testA**	**testB**	**val**	**testA**	**testB**
Sorted by size	128	26.08	29.63	22.81	23.67	26.47	21.13
	32	26.12	28.66	23.82	23.88	25.81	22.23
Random	128	27.54	31.45	24.39	24.75	27.76	22.26
	64	33.17	37.42	30.51	29.52	32.77	27.03
	32	39.79	45.31	34.04	35.70	40.28	31.28
	16	**42.66**	**47.48**	**37.51**	**36.95**	**41.42**	**32.72**
RefCOCO	128	31.84	35.84	28.69	27.67	30.93	25.37

Sorting objects by area before feeding them to the model introduces a bias, learning to recurrently segment objects in the order of their area instead of fully exploiting the referring expressions for localization. Feeding the referents as in RefCOCO does not introduce such a strong bias, but since no data augmentation strategy is used, and the number of images in the dataset is limited, the model does not learn to generalize properly. From the results in Table II, the best strategy is to randomly feed the referents, which acts as a sort of regularization which helps the model to generalize and reduces overfitting. For this configuration, smaller image batch sizes work better, a batch size of 16 images seems to be appropriate for this task and dataset.

D. Referring expressions as weak supervision

Once the best configuration regarding language encoding is selected, we can validate the performance of the referring expression branch by comparing the results with the baseline case of not having it. In this case, the model exploits the spatial recurrence property of RVOS-S to generate a sequence of masks of the same length of the amount of reference phrases associated to the image. In this task there is no supervisory signal, so the predicted masks may not follow the order of the annotations. Instead of forcing a specific order when matching the predicted masks and ground truth masks, the Hungarian algorithm [27] carries out an optimal assignment between them using the soft Intersection over Union score as cost function. This corresponds exactly to the RVOS-S zero-shot (unsupervised) setup described in [4].

Table III compares the results obtained with RVOS-S and those with our model with linguistic references. In each pair of comparable configurations, our proposed model consistently outperforms the configuration without referring expression. These results clearly indicate how the linguistic phrases are indicating which instance to segment, and that actually also help in the quality of the masks compared to the case with the zero-shot RVOS.

E. Comparison with state-of-the-art

Table IV presents comparisons of our method with existing state-of-the-art approaches. The table indicates that the quality of the segmentations still has room for improvement. Nevertheless, it must be understood that this master thesis is a first step into an end-to-end video object segmentation engine with linguistic reference expressions as the only way of supervision. The

Table III: Results in RefCOCO with and without referring expressions.

Referent order	Batch size	Referring expression	Instance IoU			Overall IoU		
			val	**testA**	**testB**	**val**	**testA**	**testB**
Sorted by size	128		21.82	25.56	18.86	18.48	21.27	16.48
	128	✓	26.08	29.63	22.81	23.67	26.47	21.13
	32		21.68	23.50	19.67	19.42	21.02	17.94
	32	✓	26.12	28.66	23.82	23.88	25.81	22.23
Random	128		20.36	22.70	15.78	17.65	19.32	15.22
	128	✓	27.54	31.45	24.39	24.75	27.76	22.26
	32		20.13	23.13	19.04	17.77	19.83	17.24
	32	✓	39.79	45.31	34.04	35.70	40.28	31.28

present work provides a baseline to improve upon in a clean architecture added to a very fast engine such as RVOS, which is state of the art at inference time for video object segmentation.

Table IV: Comparison with the state-of-the-art in precision@X and overall IoU performance across RefCOCO dataset .

Method	**split**	**prec@0.5**	**prec@0.6**	**prec@0.7**	**prec@0.8**	**prec@0.9**	**Overall IoU**
RMI [13]	val	42.99	33.24	22.75	12.11	2.23	45.18
	testA	-	-	-	-	-	45.69
	testB	-	-	-	-	-	45.57
DMN [14]	val	-	-	-	-	-	49.78
	testA	65.83	57.82	46.80	27.64	5.12	54.83
	testB	-	-	-	-	-	45.13
RNN [2]	val	60.19	50.19	38.32	23.87	5.66	54.26
	testA	-	-	-	-	-	56.21
	testB	-	-	-	-	-	52.71
CMSA [16]	val	66.44	59.70	50.77	35.52	10.96	58.32
	testA	-	-	-	-	-	60.61
	testB	-	-	-	-	-	55.09
Ours	val	38.59	30.39	22.00	12.28	2.75	36.95
	testA	43.61	34.56	25.90	14.96	2.85	41.42
	testB	31.01	23.55	16.76	10.52	3.34	32.72

Differences between testA and testB are also reflected in the state-of-art methods, where testA tends to obtain higher IoU. Since people is the dominant class in RefCOCO dataset, more data of this category is available to the model for training. Therefore, testA, which mainly includes object belonging that class, is easier to segment.

F. Qualitative results

Figure 10 shows some qualitative results generated by our network with examples on images from RefCOCO split testA, which is focus on multiple people. Similarly, Figure 11, shows some qualitative results for split testB. The depicted results are among the good predictions of the algorithm and show how our model can distinguish between different instances of the same class. Finally, in Figure 12 it can be seen that the order of the objects ids is consistent with the order of the referring expressions, showing that the predicted order has not been guessed by chance.

While there is still room for progress, these results indicate the potential of the proposed the task with the proposed deep neural architecture.

V. TOWARDS VIDEO

Taking advantage from advances of language grounding models that use referring expressions for image segmentation, we complete this thesis with a first step towards solving the video instance segmentation task with linguistic referring expressions. The contents of this section were peer reviewed and accepted for presentation as poster and publication in the ACM Multimedia 2019 Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA) [9].

Our study proposes a simple video object grounding method to establish a baseline for future experiments. The solution, illustrated in Section V, combines two existing models off-the-shelf: (i) MAttNet [8], a language grounding model designed for images only, which uses referring expressions to obtain segmentations in an image, and (ii) RVOS [4] one-shot, a semi-supervised video object segmentation model, which takes as input videos and first frame masks to segment objects along all the video sequence.

The output of MAttNet is used as a guidance for segmentation of the referents in each video frame in the following way:

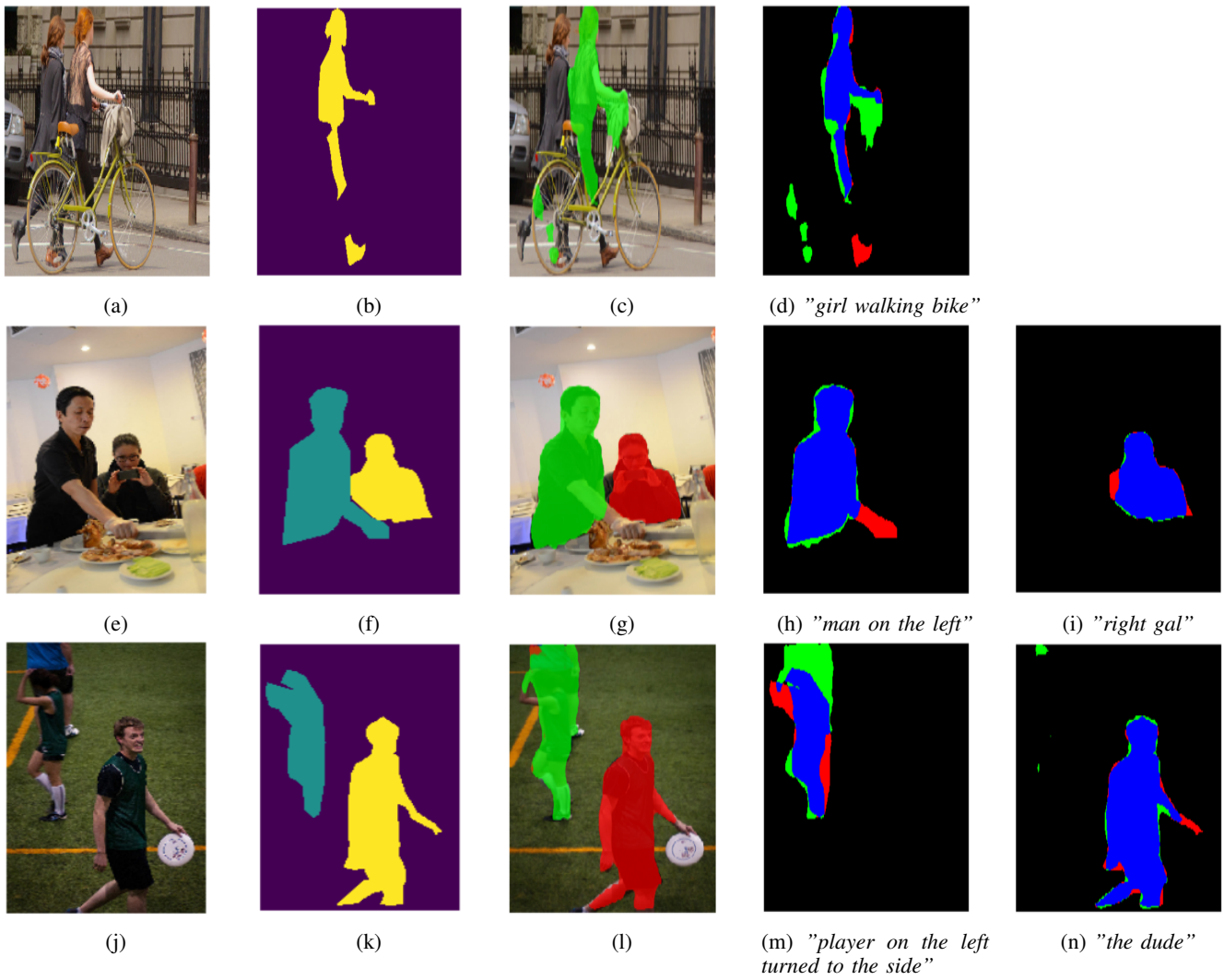


Figure 10: In RefCOCO testA, from left to right: original image, ground truth, segmentation result, visualization of the results, where red pixels are false negatives, blue true positives, green false positives and black true negatives.

1) The first video frame and a referring expression are used as input to MAttNet, obtaining a pixel-level mask of the referent in the first frame. In multi-object segmentation, this first step is applied independently for each referent, then the results are combined in a single pixel-level mask with unique instance ids. Since only one instance id can be assigned to each pixel, if there is an overlap the ID of the last object segmented is kept. Note that applying off-the-shelf image-based models to each frame independently gives temporally inconsistent results, so it is only used on the first frame.
2) The mask of combined instances for the first frame, obtained in the previous step, and the rest of the video frames are used as inputs to RVOS one-shot. The model will use the mask as a guide to segment the target objects the rest of the frames in the sequence. The model's recurrent module processes the masks obtained in previous time steps as inputs, which enforces temporal consistency of the segmentation results.

A. Experimental Setup

1) Dataset: For video object grounding, we used the Densely Annotated Video Segmentation (DAVIS) 2017 [20] dataset. It comprises videos with multiple objects densely annotated, pixel-accurate and per-frame ground truth segmentations. It consists of a training set with 60 videos, and a validation/test-dev/test-challenge set with 30 sequences each. It is a challenging dataset with complex and crowded scenes, similar looking instances, occlusions, camera view changes, fast motion, distractors, small objects, and fine structures.

All objects annotated have been augmented with non-ambiguous referring expressions provided by [7], where two non-computer vision experts annotated each referent in the dataset with a referring expression. A set of referring expressions is obtained by looking at the first frame only and a second one using the whole video sequence.

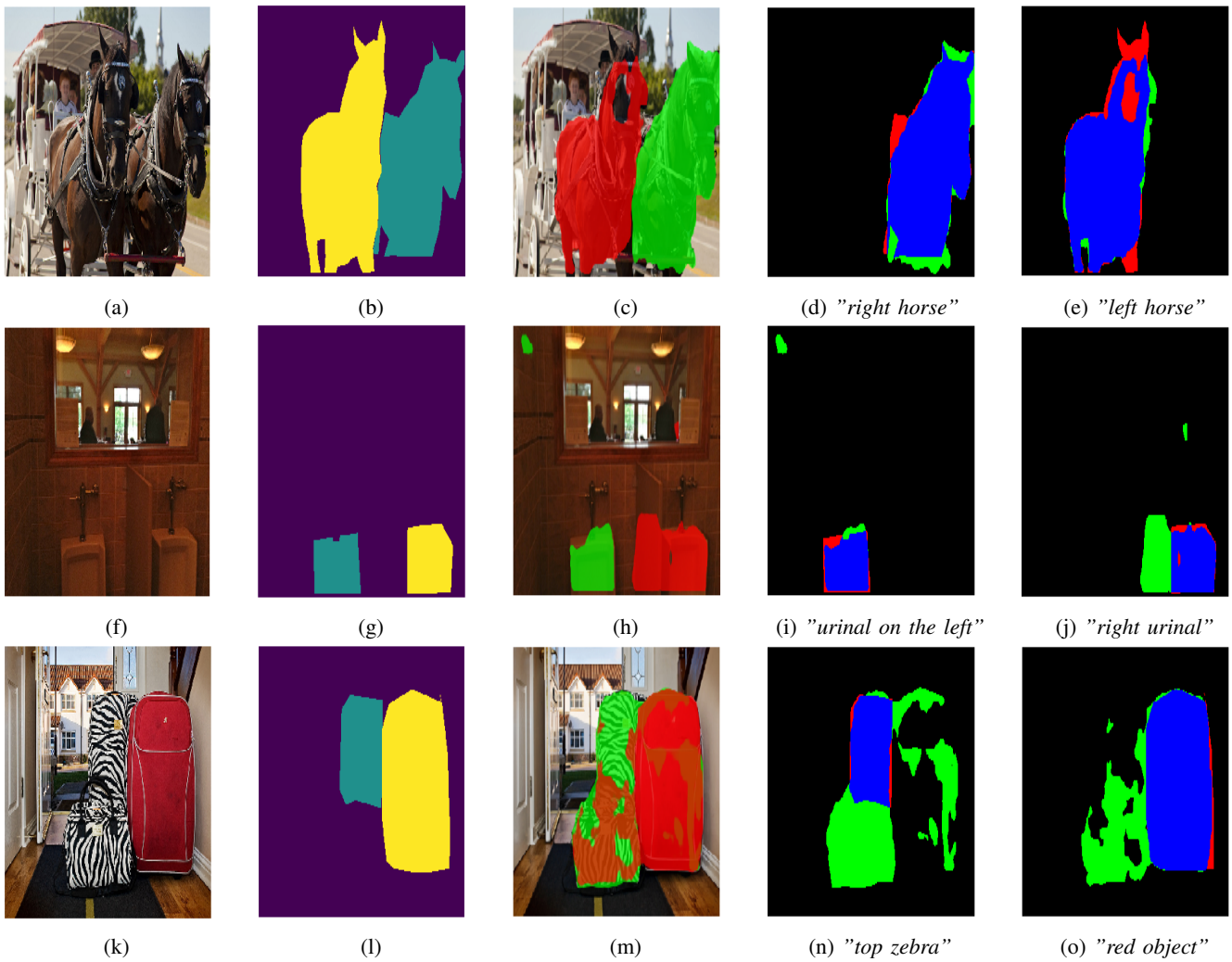


Figure 11: In RefCOCO testB, from left to right: original image, ground truth, segmentation result, visualization of the results, where red pixels are false negatives, blue true positives, green false positives and black true negatives.

Table V: Video Object Segmentation results on DAVIS_{17} Test-dev.

Supervision	**Method**	$J\&F$
First frame mask	RVOS [4]	50.3
Clicks	Scribble-OSVOS [19]	39.9
Language	MAttNet + RVOS	21.9

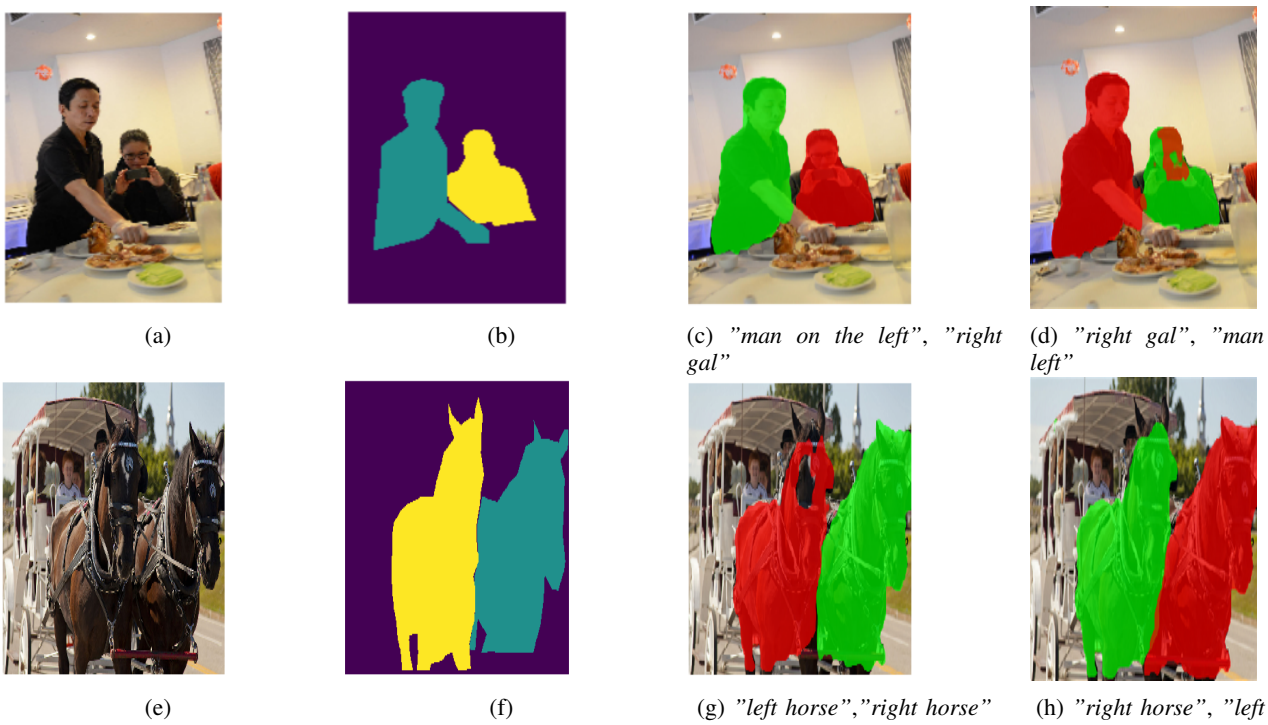
2) Metrics: For evaluation, given a predicted mask and the ground-truth mask of that same object in a frame, we use the region (Jaccard index - J) and boundary (F) measures proposed in DAVIS 2016 [28]. Jaccard index is computed as the Intersection over Union between the ground truth and the predicted segmentation averaged across all frames and video sequences. F is a measure that evaluates the contour accuracy. The greater F measure, the better the information about the contours of the predicted object has been preserved. More details about how F is computed can be found in [20]

As an overall measure of the performance, we employ the measure proposed in [20] which computes the mean of the measures $J\&F$ over all object instances.

B. Evaluation

Table V shows our video object segmentation results on the dataset DAVIS 2017 [20] with unambiguous linguistic referring expressions [7]. We show that our language-supervised approach on video significantly under-performs compared to mask semi-supervised methods due to the unstable behaviour of MAttNet, the off-the-shelf grounding model used.

In DAVIS 2017 dataset all videos contain multiple objects, hence MAttNet processes the first frame several times independently, using a referring expression at a time, to obtain a mask for each of them. When combining those into a single mask,



(a) (b) (c) *"man on the left", "right gal"* (d) *"right gal", "man on the left"*

(e) (f) (g) *"left horse","right horse"* (h) *"right horse", "left horse"*

Figure 12: Examples of results obtained changing the order the referring expressions are fed to the model. From left to right: original image, ground truth, segmentation result (original order), and segmentation results (inverse order).

every pixel is assigned to a single instance id. Therefore, wrongly localized objects might occlude segmentations belonging to correctly identified objects. This leads to the impossibility of retrieving the correct instances by the semi-supervised method. The evaluation metric strongly penalizes cases in which the id of the segmented objects does not match the annotations provided (as it can be seen in Figure 14), which severely affects the results.

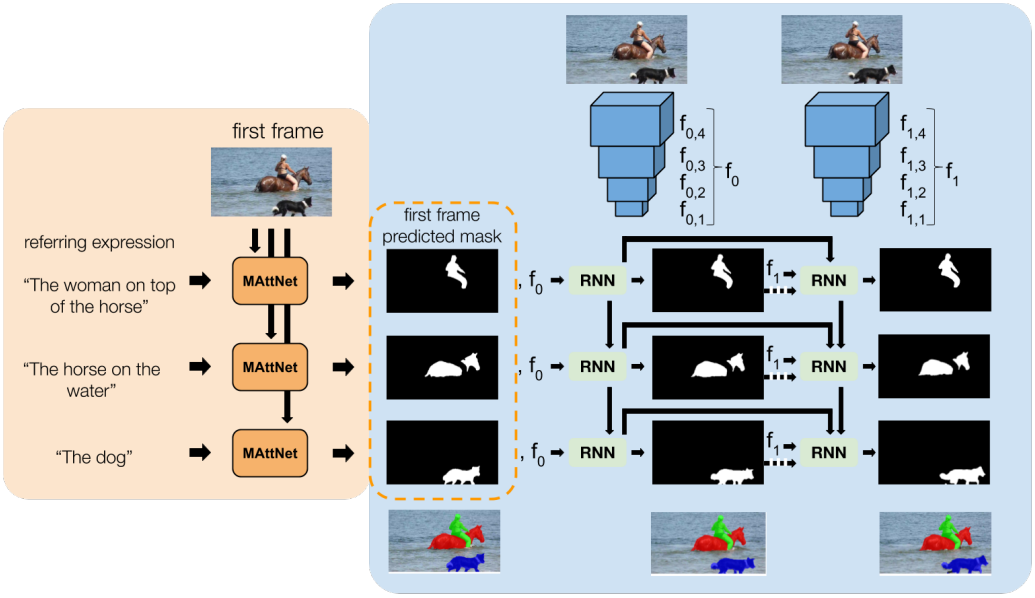


Figure 13: Video baseline. On the left, the grounding model MAttNet obtains segmentation masks of instances in the first frame. On the right, RVOS for one-shot (unsupervised) segmentation uses the masks from the first frame to consistently segment the objects along the sequence.

(a) Correct id assignment (b) Wrong id assignment

Figure 14: Comparison between masks obtained by MAttNet on the first frame of DAVIS 2017 sequences. For (a) and (b), ground truth annotation on the left and segmentation obtained on the right.

VI. Future Work

The next step of this project is to exploit RVOS full architecture, introducing the temporal recurrence to develop a fully end-to-end trainable model for multiple objects in video object segmentation using referring expressions. To the best of our knowledge, this would be the first neural architecture that solves this task in an end-to-end fashion.

We will also focus on researching novel strategies to combine linguistic and visual representations for instance segmentation in both image and video. Approaches that we consider are the fusion of embeddings at different resolutions with varying dimensions in the architecture, or training joint embedding representations to be exploited by the encoder-decoder model. We are also curious to understand why the linear projection did not outperform the PCA, as many existing works report results in the opposite direction.

Finally, more accurate representations of referring expressions can be obtained by fine-tuning the pre-trained BERT base model to perform the new task. While in the current implementation we kept the language branch frozen, results are expected to improve if these layers are trained with the RefCOCO dataset as well.

VII. Conclusions

This master thesis has proposed a solution for instance segmentation on still images by adding a new linguistic branch to the existing RVOS-S model. The concatenation of the BERT reference embeddings to each pixel embedding of the RVOS-S decoder allows selecting which instance to segment in a weakly supervised scenario. Our work has explored different variations of the BERT encodings, batch sizes or criteria to sort the referring expressions during training. While our quantitative results do not exceed the state of the art in terms of accuracy, the proposed architecture is trained end-to-end and accomplishes its main goal of preparing the path towards solving the task for videos. In this sense, the thesis also contains a baseline configuration for video object segmentation with linguistic reference expressions that will be published in an international workshop in ACM Multimedia 2019.

Finally, we would like to highlight that this master thesis follows a global trend of solving multimodal tasks with deep neural networks. The popularity of these machine learning tools in both the computer vision and natural language communities, has opened new and exciting research directions. In the very near future, many multimedia practitioners will not only embrace a single modality, but feel confident with multiple ones, such as language and vision as in this thesis, but also, for example, audio, physiological sensors or any type of medical records.

Acknowledgment

I would like to express my gratitude to the supervisors of this project, Carles Ventura, Xavier Giró and Carina Helga Silberer, for their continuous support, encouragement and dedication. Special thanks to the COLT (Computational Linguistics and Linguistics Theory) group for their collaboration, specially to Gemma Boleda and Ionut-Teodor Sorodoc. Also, thanks to the Image Processing Group of the Universitat Politècnica de Catalunya for providing all the necessary resources.

References

[1] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions", in *European Conference on Computer Vision*, Springer, 2016, pp. 108–124.

[2] R. Li, K. Li, Y.-C. Kuo, M. Shu, X. Qi, X. Shen, and J. Jia, "Referring image segmentation via recurrent refinement networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5745–5753.

[3] H. Shi, H. Li, F. Meng, and Q. Wu, "Key-word-aware network for referring expression image segmentation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 38–54.

[4] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i Nieto, "Rvos: End-to-end recurrent network for video object segmentation", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[6] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions", in *European Conference on Computer Vision*, Springer, 2016, pp. 69–85.

[7] A. Khoreva, A. Rohrbach, and B. Schiele, "Video object segmentation with language referring expressions", in *Asian Conference on Computer Vision (ACCV)*, 2018.

[8] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension", in *CVPR*, 2018.

[9] A. Herrera-Palacio, C. Ventura, and X. Giro-i Nieto, "Video object linguistic grounding", in *Proceedings of the first ACM international workshop on Multimodal Understanding and Learning for Embodied Applications*, ACM, 2019.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality", in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations", *arXiv preprint arXiv:1802.05365*, 2018.

[13] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille, "Recurrent multimodal interaction for referring image segmentation", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1271–1280.

[14] E. Margffoy-Tuay, J. C. Pérez, E. Botero, and P. Arbeláez, "Dynamic multimodal instance segmentation guided by natural language queries", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 630–645.

[15] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting", in *Advances in neural information processing systems*, 2015, pp. 802–810.

[16] L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 502–10 511.

[17] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[18] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 616–625.

[19] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset, "The 2018 davis challenge on video object segmentation", *arXiv preprint arXiv:1803.00557*, 2018.

[20] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation", *arXiv preprint arXiv:1704.00675*, 2017.

[21] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", in *CVPR09*, 2009.

[24] A. Salvador, M. Bellver, V. Campos, M. Baradad, F. Marques, J. Torres, and X. Giro-i Nieto, "Recurrent neural networks for semantic instance segmentation", *arXiv preprint arXiv:1712.00617*, 2017.

[25] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes", in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context", in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[27] H. W. Kuhn, "The hungarian method for the assignment problem", *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[28] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.