University of Padua, Department of Mathematics "Tullio Levi-Civita"

PH.D. IN BRAIN, MIND AND COMPUTER SCIENCE
Curriculum in Computer Science and Innovation for Societal Challenges
XXXV series

Understanding Multimedia Content with Prior Knowledge

Ph.D. Candidate:
Davide Rigoni

Supervisor:
Prof. Luciano Serafini

Co-Supervisor:
Prof. Alessandro Sperduti

Ph.D. Course Coordinator:
Prof. Anna Spagnolli

Years 2019-2023

This Ph.D. thesis is dedicated to my family:
my parents *Giovanni Battista Rigoni* and *Susanna Cunico*;
my brother *Giulio Rigoni*;
my grandmothers *Francesca Covolo* and *Caterina Cunico*.

Abstract

Visual-textual grounding is a challenging task that involves associating language with visual objects or scenes, and it has become a popular research area due to its importance in various applications. Traditionally, visual-textual grounding has been solved by relying on information from images and textual phrases. However, incorporating additional prior knowledge, such as a graph, could potentially enhance the performance and accuracy of visual-textual grounding models. The graph is a discrete structure that can represent any kind of information that can be used to solve the grounding task.

In this Ph.D. thesis, a formal probabilistic framework is proposed to consider all three modalities: image, text, and graph. The framework allows for the analysis of existing works and the development of a novel approach to visual-textual grounding based on an innovative factorization of probabilities. The adoption of the probabilistic approach is crucial for accounting for the inherent uncertainties in solving the task.

In addition, this thesis presents two contributions to improve the traditional visual-textual grounding task. The first contribution regards a new loss function for training visual-textual grounding models in a supervised setting. Indeed, the models in the literature are typically constituted by two main components that focus on how to learn useful multi-modal features for grounding and how to improve the predicted bounding box of the visual mention, respectively. Finding the right learning balance between these two sub-tasks is not easy, and the current models are not necessarily optimal with respect to this issue.

The second contribution consists of a model tackling the weakly-supervised visual-textual grounding. The proposed model is based on the principle of first predicting a rough alignment among phrases and boxes, adopting a module that does not require training, and then refining those alignments using a learnable neural network. The model is trained to maximize the multimodal similarity between an image and a sentence describing that image while minimizing the multimodal similarity of the same sentence and a new unrelated image, carefully selected so as to help as much as possible during training.

The object detector plays a fundamental role in solving the visual-textual grounding task. It should be able to identify many different objects and classify them correctly. Nevertheless, increasing the number of objects to be recognized usually leads to a more challenging classification problem. The importance of the correct classification of an object is even greater when considering the graph in the resolution of the visual-textual grounding task. In fact, the semantic information conveyed through the classes is crucial to identify the graph nodes that best characterize the objects depicted in the image. In literature, the most common approach is to use an object detector trained to detect 1600 different classes of objects. However, those classes are noisy and impair the performance of the object detector. To solve this problem,

this document proposes also a new set of clean labels to use for training object detectors on the Visual Genome dataset.

To conclude, this thesis introduces a new object detector that can be conditioned by nodes of the WordNet graph to search for objects in images. In particular, the conditioned object detector can be deployed to estimate a component of the probability distribution factorization designed thanks to the probabilistic framework.

Overall, this Ph.D. thesis contributes to the study of visual-textual grounding and provides tools and insights that have the potential for developing advanced approaches and applications within this domain.

Acknowledgement

Not surprisingly, I would like to thank my parents, *Giovanni Battista Rigoni* and *Susanna Cunico*, for their unwavering support and belief in me. Their sacrifices, understanding, and encouragement have made this journey possible and worthwhile. I am also grateful to my brother *Giulio Rigoni*, for the help, suggestions, and all the feedback he has given me along this path.

I want to express my deepest gratitude to my supervisors, *Alessandro Sperduti* and *Luciano Serafini*, whose guidance and support have been invaluable throughout these years of research. They have inspired and provided me with precious ideas. Furthermore, I thank the Bruno Kessler Foundation for funding the Ph.D. scholarship and Prof. *Desmond Elliott* for hosting me under his guidance at the University of Copenhagen.

Last but not least, I owe a special thanks to my partner *Rossella Goffredo*, for the unlimited source of advice and support in every moment of need, to whom I owe more than words can express.

Padova, 31 May 2023
– Davide Rigoni

Ringraziamenti

Non sorprendentemente, vorrei ringraziare i miei genitori, *Giovanni Battista Rigoni* e *Susanna Cunico*, per il loro incrollabile sostegno e fiducia in me. I loro sacrifici, la loro comprensione e il loro incoraggiamento hanno reso possibile e utile questo viaggio. Sono anche grato a mio fratello *Giulio Rigoni* per l'aiuto, i suggerimenti e tutti i feedbacks che mi ha dato durante questo percorso.

Desidero esprimere la mia più profonda gratitudine ai miei supervisori, *Alessandro Sperduti* e *Luciano Serafini*, per la loro preziosa guida e supporto durante questi anni di ricerca. Sono stati fonte di ispirazione e mi hanno fornito spunti preziosi. Inoltre, desidero ringraziare la Fondazione Bruno Kessler per aver finanziato la borsa di dottorato e il Prof. *Desmond Elliott* per avermi ospitato sotto la sua guida all'Università di Copenhagen.

Ultimo ma non meno importante, devo un ringraziamento speciale alla mia compagna *Rossella Goffredo*, per la fonte illimitata di consigli e supporto in ogni momento di bisogno, a cui devo più di quanto le parole possano esprimere.

Padova, 31 May 2023
– Davide Rigoni

Contents

Listing of figures

Listing of tables

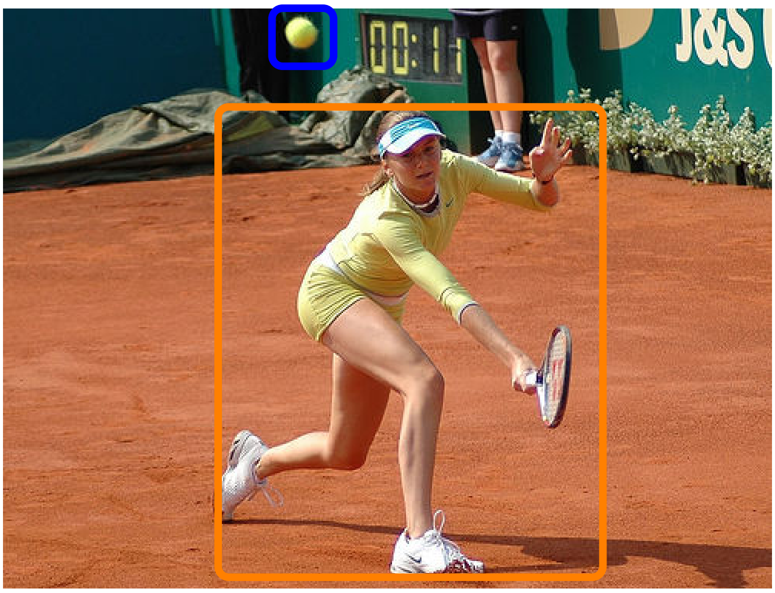
1 Introduction

According to Zou *et al.* [5], the number of computer vision researches published by year increased from less than 100 in 1998 to slightly almost 3500 in 2021. Under this wave of increasing interest from the research community, several ideas and approaches were specifically designed to solve vision and natural language problems, such as visual-textual grounding [6, 7, 8, 9, 10, 11], visual question answering [12, 13, 14], visual-textual-knowledge entity linking [15, 16, 17] and image-text retrieval [18, 19, 20, 21, 22].

The causes for these remarkable fast developments are mainly the following: *(i)* the continuous development of hardware that allows fast computations enables the use of complex models capable of capturing detailed behavior; and *(ii)* the continuous availability of new open-source datasets. However, comprehending both visual and textual modalities remains nowadays a difficult task.

Among all the visual and language research areas, the research community has devoted much effort to solve the visual-textual grounding task, also known as referring expression grounding. The visual-textual grounding is a task in computer vision and natural language processing that involves associating language with visual objects or scenes. It aims to ground language in the visual world by mapping words or phrases in a sentence to specific objects or regions in an image or video.

For example, given the image in Figure 1.1 and the sentence “A woman tries to volley a tennis ball”, the visual-textual grounding task would involve identifying the location of the woman in the image and highlighting the corresponding region of the tennis ball. This re-



Sentence: "A woman tries to volley a tennis ball".

Figure 1.1: Visual-textual grounding example given the textual phrase "A woman tries to volley a tennis ball". The word "woman" refers to the rectangle in orange, while the words "tennis ball" refer to the rectangle in blue.

quires the system to understand the meaning of the sentence and to use visual cues to identify the object in the image that corresponds to the word "woman" (i.e., rectangle in orange) and the region that corresponds to the word "tennis ball" (i.e., rectangle in blue). Usually, the region in the image is delimited by a square named "bounding box", and the part of the sentence that refers to it is called "query". The textual phrase "volley a tennis ball" could also be grounded to the corresponding region of the image depicting the racket. In this case, the textual phrase involves the action performed by the woman, which implicitly refers to the racket depicted in the image used to volley the tennis ball. Adopting a similar reasoning, also the "volley" word could be grounded to the racket in the image, albeit in this case, the implicit joint reasoning between image and text requires the understanding that the action "volley" refers to the tennis ball. A general knowledge of the world is required to perform this implicit grounding of the action. This document addresses the standard visual-textual grounding problem, which grounds the noun phrases to the corresponding objects depicted in the image, and it does not cover the aspect of grounding actions.

The visual-textual grounding task is important in many applications, such as image and video captioning, visual question answering, and robotics, where machines must interact with the physical world and understand natural language commands or descriptions. It is a challenging task because it requires both language understanding and visual perception and often involves dealing with noisy and ambiguous data.

This document delves into the problem of visual-textual grounding. In the literature, there are mainly two kinds of approaches to solve this problem, namely "two-stage" and "one-stage" approaches. In the two-stage approach, the visual-textual grounding task is cast as a sequence of two sub-tasks: an object detection task followed by a classification task. The object detection task aims to find all the objects depicted in the image, while the visual-textual grounding model, given the textual phrase, returns only the detected object in the image that represents the best semantic match with the sentence. In the initial phase of research on this problem, many works have followed this formulation, developing several approaches [6, 23], while more recent works have chosen to address the problem by a "one-stage" approach model, in which the object detection and the classification problem are solved jointly [24, 25].

In the "two-stage" approach, the visual-textual grounding model receives in input a set of proposal bounding boxes previously extracted by an object proposals extractor, such as Edge Boxes [26] and Selective Search [27], or by an object detector, such as Faster R-CNN [28], Single Shot multibox Detector (SSD) [29], or YOLO [30, 31]. These proposals, jointly with the given input textual sentence describing the content of the image, constitute the visual-textual grounding model input. Usually, the model embeds the sentence in an embedding representation that tries to capture its semantic content. Then, the model predicts, for each proposal bounding box, a score that represents how much the content of the bounding box is likely to be referred by the sentence. Often, the two-stage approach models predict new coordinates for the best-predicted proposal in order to adjust the coordinates to better fit the visual content according to the sentence semantic information.

In the one-stage approach, the visual-textual grounding model receives only an image and a textual sentence in input. Then the model learns how to extract and fuse all the visual and textual information to predict the best bounding box in output, according to the input sentence. Even if this seems to be the best approach in order to reach the best results, due to the small number of assumptions made by the model, it raises some major issues: *(i)* not all the visual-textual grounding datasets are suitable for training an object detector due to lack of images and/or because they are not densely annotated; *(ii)* the model requires a high number of parameters, and because of that *(iii)* the training requires significant computing resources.

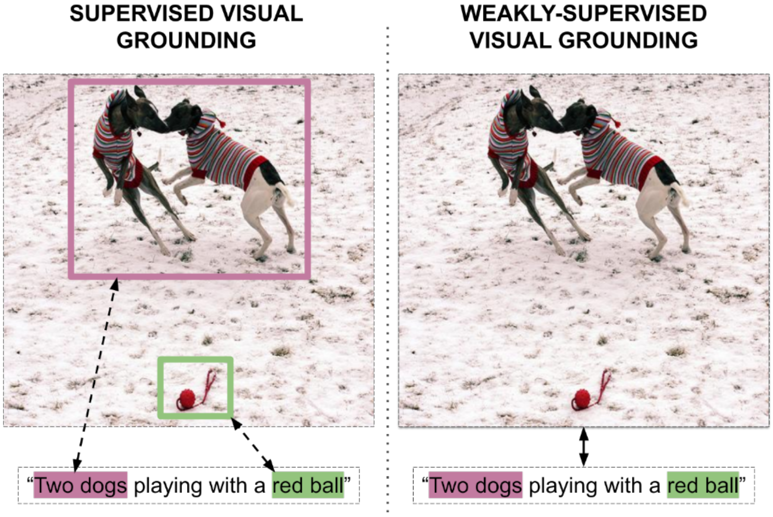

SUPERVISED VISUAL GROUNDING
WEAKLY-SUPERVISED VISUAL GROUNDING
“Two dogs playing with a red ball”
“Two dogs playing with a red ball”

Figure 1.2: Differences between the supervised visual-textual grounding and the weakly-supervised visual-textual grounding task. On the left, the standard visual-textual grounding task is presented, where the dataset annotations contain the link between queries and boxes. In contrast, on the right, the only available annotation is the information that links a description with its own image and vice-versa.

According to the literature, it is also possible to group the models according to the level of supervision available during the model training phase. In the supervised category, models are trained using all the region-phrase pairs [6, 7, 8, 9, 10, 11]. Figure 1.2, on the left, reports and examples of all the annotations needed during training, which consist of: *(i)* bounding boxes coordinates; *(ii)* textual phrases; and *(iii)* region-phrase matching.

In the weakly-supervised category, models during training do not use the whole information available from the visual-textual grounding dataset. More in particular, during the training phase, the model is only given to know that a given textual phrase refers to some objects depicted in an image. However, the model does not have access to the bounding box coordinates or the region-phase match. Figure 1.2, on the right, presents the annotations available under the weakly-supervised setting. In general, given the less information available during model training, weakly-supervised approaches perform less than supervised ones.

Researchers have traditionally solved the visual-textual grounding problem by relying on information from bounding boxes and textual queries. However, appropriately integrating additional prior knowledge could potentially enhance the performance and accuracy of visual-textual grounding models. In this context, this Ph.D. thesis aims to augment the conventional approach to solve this task. While the classic method involves two input modalities (text and image), this document proposes incorporating a third modality in the form of a graph. The graph is a discrete structure that can represent any kind of information that can be used to solve the grounding task.

The addition of the graph requires the reconciliation of the information conveyed through the two modalities (i.e., image and text) with the information conveyed with the graph modality. Thus, the model should align the graph's nodes with the bounding boxes and textual queries to solve the visual-textual grounding task.

To analyze the introduction of the new modality, this document proposes a formal probabilistic framework designed to consider all three modalities: image, text, and graph. The adoption of the probabilistic approach is crucial for accounting for the inherent uncertainties in solving the visual-textual grounding task. The framework allows the analysis of the already published works, highlighting their strengths and weaknesses according to how the modalities are adopted in the model.

The framework constitutes an important tool that can be employed to devise a novel approach to visual-textual grounding based on an innovative factorization of probabilities not yet explored in the literature. Indeed, in this Ph.D. thesis, a new factorization of the distribution with an estimation of each component will be proposed.

During the development of the proposed framework, the traditional visual-textual grounding task based on two modalities (i.e., image and text) was also studied for improvements. In this direction, two contributions have been proposed. The first proposes a new loss for training two-stage models in the supervised setting, while the latter contribution proposes a two-stage model to solve the visual-textual grounding task in the weakly-supervised setting.

In the visual-textual grounding, particularly in the two-stage approach, it is evident that the bounding boxes detected with the object detector play a fundamental role in solving the problem. For this reason, the object detector should be able to identify and classify many different objects correctly. Nevertheless, the increase in the number of objects to be recognized usually leads to a more challenging classification problem. The importance of the correct classification of an object is even greater when considering the graph in the resolution of the visual-textual grounding task. In fact, the semantic information conveyed through the classes is crucial to identify the graph nodes that best characterize the objects depicted in the image.

The Bottom-Up Faster R-CNN [28] (BUA) object detector is the most commonly used by the multimodal language-and-vision community because it is trained on a pre-defined set of 1600 classes. Although the high number of classes allows the detection of many different types of objects often referred by the textual phrase, the pre-defined set of class labels is very noisy. These noisy labels may result in a sub-optimal representational space and likely impair the ability of the model to classify objects correctly. For this reason, in this Ph.D. thesis, a set of less noisy labels is also proposed.

Overall, this Ph.D. thesis contributes to the study of visual-textual grounding and provides tools and insights that have the potential for developing advanced approaches and applications within this domain.

The document is structured as follows:

- Chapter 2 introduces background information essential to understand the ideas and concepts presented in this document;
- Chapter 3 presents the visual-textual grounding State-of-the-Art;
- Chapter 4 formally presents the visual-textual grounding problem, the probabilistic framework, and an innovative proposal to solve the visual-textual grounding task;
- Chapter 5 introduces a model that solves the standard visual-textual grounding task in a supervised setting;
- Chapter 6 introduces a model that solves the standard visual-textual grounding task in a weakly-supervised setting;
- Chapter 7 introduces two potential extensions of the presented approaches. In particular, Section 7.1 delves into the problem of object detectors proposing a set of cleaned labels to adopt when training the object detectors on the Visual Genome [32] dataset. Instead, Section 7.2 presents a new object detector that can be conditioned by nodes of the WordNet [33] graph to search the objects in the images. In particular, this approach estimates a component of the new factorization presented in Chapter 4.
- Chapter 8 concludes the document and presents future works.

2 Background

This chapter will review some necessary background material concerning fundamental concepts necessary to contextualize and understand the ideas presented in this manuscript. The material will focus solely on the subset of notions required to understand most of the discussion in the rest of the dissertation.

2.1 Notation

In order to explain our work, the following notation will be used: *(i)* lower case symbols for scalars, indexes, and assignation to random variables, e.g., n and x; *(ii)* italics upper case symbols for sets and random variables, e.g., A and X; *(iii)* upper case symbols for textual sentences, e.g., S; *(iv)* bold lower case symbols for vectors and assignations to vectors of random variables, e.g., $\boldsymbol{a}$ and $\boldsymbol{x}$; *(v)* bold upper case symbols for matrices, tensors, and vectors of random variables, e.g., $\boldsymbol{A}$ and $\boldsymbol{Z}$; *(vi)* the position within a tensor or vector is indicated with numeric subscripts, e.g., $\boldsymbol{A}_{ij}$ with $i, j \in \mathbb{N}^+$; *(vii)* calligraphic symbols for domains, e.g., $\mathcal{Q}$.

2.2 Word Embeddings

Words are discrete entities that allow one to formulate textual phrases. However, neural networks that deal with natural language need continuous representations to learn and solve many natural language processing tasks, such as information retrieval [34], document classification [35], question answering [36], named entity recognition [37], and parsing [38]. Instead of treating words as atomic units where there is no notion of similarity between words, as these are represented as indices in a vocabulary, natural language information can be represented as feature vectors in a semantic space. In this space, words are associated with continuous real-valued vectors of fixed dimension, and words with semantically similar meanings tend to have similar representations. More specifically, word embedding can be defined as a learned representation of text where words that have the same meaning have a similar representation. In the literature, there are several approaches for learning good word embeddings. Some use machine learning [39, 40, 3] techniques while others use statistical approaches like Latent Semantic Analysis (LSA) [41]. In the following, some famous and successful word embeddings will be described.

2.2.1 Word2Vec and GloVe

Word2Vec, introduced by T. Mikolov in [39] and then updated in [40, 42], is a statistical method for efficiently learning word embeddings from a text corpus. Word2Vec aims to learn meaningful representations of words that capture semantic information. Two different approaches are used: Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram Model. The first learns embeddings by predicting the current word based on its context, while the latter learns by predicting the surrounding words given a current word. The context is defined by a window of neighboring words and can be configured or fine-tuned. The quality of these representations is measured in a word similarity task. The key benefit of the approach is that high-quality word embeddings can be learned using low space and time complexity, allowing larger embeddings to be learned from much larger corpora of text.

GloVe [3] is another approach to learning word embeddings. The main goal of GloVe is to overcome Word2Vec problems related to the lack of statistical information. Rather than using a window to define local context, GloVe constructs an explicit word co-occurrence matrix using statistics across the whole text corpus. The result is a learning model that may result in generally better word embeddings.

2.2.2 BERT Embeddings

Bidirectional Encoder Representations from Transformers (BERT) [43] is a language representation model, designed to pre-train deep bidirectional word representations from the unlabeled text. BERT uses deep transformer architecture instead of a traditional recurrent neural network, and it has many differences concerning traditional word embeddings. Differently from Word2Vec or GloVe, it produces multiple embeddings for the same word based on the context in which the word is used. For example, in the sentences "The head of the body" and "The head of the department", the word "head" produces different embeddings in BERT for each sentence, while in either Word2Vec or Glove, such word is represented by a single, unique vector. Another difference is given by the fact that BERT explicitly leverages information involving the position of the word in the sentence, which enhances the context dependence. This implies that the model input should be a sentence rather than a single word. BERT can also be used in single-word mode (as Word2Vec and GloVe), but this breaks the advantage of generating context-dependent embeddings.

2.3 NLP and Part of Speech Tagging

Part of Speech (POS) tagging is a well-known Natural Language Processing (NLP) problem that consists of assigning to each word of a textual phrase the appropriate morphosyntactic tag in its context of appearance [44]. Due to the intrinsic properties of natural languages, those tags highly depends on the context in which words appear. In most cases, words can be disambiguated entirely taking into account an adequate context, although in others disambiguation is very difficult. According to the literature [44], existing taggers can be grouped into three main categories according to the type of knowledge they use.

In the first category, there are systems that hardcode a set of rules written by linguists. The number of rules used by such systems may reach up to several thousand rules and for this reason, the model design requires a lot of time and effort.

In the second category, there are approaches relying on statistical methods to disambiguate word meanings. These models usually encode information as a set of co-occurrence frequencies estimated from the training corpus.

In the last category, there are machine learning approaches that use more complex information than co-occurrence frequencies.

POS tags were originally introduced by Marcus Mitchell *et al.* [1] in the Penn Treebank corpus, which is a comprehensive list of tags for english words. These tags include nouns, adjectives, verb tenses, and also symbols. Figure 2.1 shows the tags along with their description.

Since then the research community has devoted much attention to this task and has developed several new approaches, such as Stanford Dependencies [45, 46, 47], Google universal

CC	Coordinating conjunction	PRP	Personal pronoun	WDT	*wh*-determiner
CD	Cardinal number	PP$	Possessive pronoun	WP	*wh*-pronoun
DT	Determiner	RB	Adverb	WP$	Possessive *wh*-pronoun
EX	Existential *there*	RBR	Adverb, comparative	WRB	*wh*-adverb
FW	Foreign word	RBS	Adverb, superlative	#	Pound sign
IN	Preposition	RP	Particle	$	Dollar sign
JJ	Adjective	SYM	Symbol	.	End of sentence
JJR	Adjective, comparative	TO	*to*	,	Comma
JJS	Adjective, superlative	UH	Interjection	:	Colon, semi-colon
LS	List item marker	VB	Verb, base form	(	Left bracket character
MD	Modal	VBD	Verb, past tense	)	Right bracket character
NN	Noun, singular	VBG	Verb, gerund	"	Straight double quote
NNP	Proper noun, singular	VBN	Verb, past participle	‘	Left open single quote
NNS	Noun, plural	VBP	Verb, non-3rd ps. sing. present	‘‘	Left open double quote
NNPS	Proper noun, plural			’	Right close single quote
PDT	Predeterminer	VBZ	Verb, 3rd ps. sing. present	’’	Right close double quote
POS	Possessive ending				

Figure 2.1: Part-of-speech set of English tags of the Penn Treebank corpus [1].

part-of-speech tags [48] and the Interset interlingua [49] for morphosyntactic tagsets. Universal Dependencies (UD) [50, 2] is a recent approach that aims to achieve annotation consistency among different languages, maintaining language-specific annotation when necessary. Figure 2.2 reports new Universal POS tags.

Nowadays, thanks to the available open-source datasets and the huge interest in NLP areas, several NLP tools exist. Probably the best known of all of these is the Stanford CoreNLP [51], despite many others are also frequently adopted, like Flair [52], spaCy [53], UDPipe [54] and Stanza [55].

2.4 Object Detection and Recognition Systems

Object detection (OD) and object recognition (OR) systems are essential for many commonplace tasks, including face detection, information retrieval from image and video databases, surveillance applications, driver assistance, automation, and more in particular in computer vision, it is a fundamental building block used to extract information from images. These systems' core functionality can be divided into two parts: finding objects in a picture by, for example, drawing a bounding box around them (object detection), and then classifying the

Traditional POS	UPOS	Category
noun	NOUN	common noun
	PROPN	proper noun
verb	VERB	main verb
	AUX	auxiliary verb or other tense, aspect, or mood particle
adjective	ADJ	adjective
	DET	determiner (including article)
	NUM	numeral (cardinal)
adverb	ADV	adverb
pronoun	PRON	pronoun
preposition	ADP	adposition (preposition/postposition)
conjunction	CCONJ	coordinating conjunction
	SCONJ	subordinating conjunction
interjection	INTJ	interjection
–	PART	particle (special single word markers in some languages)
–	X	other (e.g., words in foreign language expressions)
–	SYM	non-punctuation symbol (e.g., a hash (#) or emoji)
–	PUNCT	punctuation

Figure 2.2: Universal part-of-speech tags (UPOS) [2].

objects using the classes it was taught on (object recognition). Because OD and OR are frequently combined, and typically, the name of two tasks usually collapses into one of the two depending on the relevance in the system.

Nowadays, object detector architectures are composed of two main components: the backbone and the object detector head. The former takes in input an RGB image and processes it to extract global features of the content of the image. The latter, starting from the features generated by the backbone, aims to locate and classify the objects in the image.

Many object detectors exist [29, 31, 56, 28, 57, 58, 59, 60, 61, 62, 63, 64, 56, 65], that differ according to their ability to detect objects in the image, the computing power required for their use, and their ability to recognize a large set of different objects[66, 67]. An object detector should be able to identify many different objects [23] and classify them correctly.

In the following sections, some common object detectors will be presented.

2.4.1 R-CNN, Fast R-CNN and Faster R-CNN

Region-based convolutional networks (R-CNN) [68] is one of the first object detectors ever developed. It is composed of a two-stage architecture that is not trained end-to-end. In the first step, the model deploys the Selective Search [27] algorithm to generate around 2000 category-independent region proposals for the input image, while in the last step, it classifies each region using a Support Vector Machine (SVM) [69]. In particular, the SVM takes as input a vector of features extracted with a convolution neural network from each region proposal.

The Fast R-CNN [70] model was built to address a few disadvantages of the R-CNN model, such as the two-stage approach. This model takes as input an image and a set of region proposals (ROIs) generated with Selective Search. Initially, all the RGB image is fed to a convolutional neural network (like a backbone) that extracts the image features, from which an ROI pooling layer extracts the features corresponding to each region proposals. Then, Fast R-CNN adopts two fully connected neural networks: *(i)* to classify each region proposal according to the pre-defined set of labels; and *(ii)* to refine the coordinates of each region proposal to delimit better the object in the image.

The Faster R-CNN [28, 71], unlike the previous two models, does not use the Selective Search algorithm and implements a new neural network, namely the Region Proposal network (RPN), that aims to predict the region proposals. As for the Fast R-CNN model, the proposals are fed to the ROI pooling layer that extracts feature representations for each proposal. Then, each proposal is classified and its coordinates are refined through fully connected neural networks. The model is trained end-to-end for both locating and classifying the objects. In particular, the RPN layer is trained with the only objective of locating the object in the image (i.e., binary classification).

2.4.2 Other Recent Object Detectors

You Only Look Once (YOLO) [67, 31, 56] is a State-of-the-Art object detector that provides high precision and speed. This model simultaneously detects bounding boxes and classifies them according to its pre-defined set of classes adopting convolution neural networks. It is trained to maximize its detection performances in an end-to-end manner.

Another State-of-the-Art model for object detection is Single Shot Detector (SSD) [29]. SSD divides the image using a grid at multiple resolutions and scales. Then, for each grid cell, it detects and classifies all the objects in that region of the image. SSD, similar to YOLO, can detect objects in a short amount of time.

RetinaNet [65] is another State-of-the-Art object detector that utilizes a focal loss function to address class imbalance during training. Focal loss applies a modulating term to the cross entropy loss in order to focus learning on hard negative examples. RetinaNet is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a convolutional feature map over an entire input image and is an off-the-self convolutional network, such as ResNet [72] and VGG [73]. The first subnet performs convolutional object classification on the backbone's output, while the second subnet performs convolutional bounding box regression to refine the coordinate of the region proposal. The two subnetworks feature a simple design that the authors propose specifically for one-stage detection.

2.5 WordNet

WordNet [33] is a large open-source lexical database of English words and their semantic relationships. It was created at Princeton University and was first released in 1985. It contains more than 155000 words and more than 117000 synsets (sets of synonyms) organized into a network of hierarchies, with each synset representing a distinct concept. WordNet has been used in various applications, including search engines, question-answering systems, and machine-learning algorithms. It has also been extended to include languages besides English, such as Spanish and Italian.

The database is structured in a way that allows words to be grouped together based on their meanings and the relationships between them, such as synonyms, antonyms, hypernyms (more general terms), hyponyms (more specific terms), and meronyms (parts of a whole). This makes it a valuable resource for natural language processing tasks, such as text classification, information retrieval, and machine translation.

WordNet was created through a combination of manual effort and automated techniques. The initial version of WordNet was developed by a team of researchers led by George A. Miller, who manually identified and organized sets of synonyms, antonyms, and other semantic relationships for a large number of words. To create WordNet, the researchers used a method called "lexical sampling", which involved selecting a small set of words from various semantic domains (such as animals, plants, and household objects) and then identifying and organizing their semantic relationships. This initial set of words and their relationships formed the basis for expanding the database to include more words and concepts. As WordNet grew, automated techniques were developed to help identify and organize the semantic relationships between words. These techniques included natural language processing algorithms that could analyze large volumes of text to identify patterns in how words are used together, as well as algorithms for clustering and grouping related words.

Over time, WordNet has been continually refined and expanded, with new words and relationships added to the database based on ongoing research and feedback from users.

3 Related Works

This chapter reviews the literature and State-of-the-Art approaches adopted to solve the visual-textual grounding task. Initially, the presentation will cover approaches adopting the fully-supervised setting, while subsequently, it will cover approaches working on the weakly-supervised setting. Then, the Visual-Textual-Knowledge Entity Linking (VTKEL) problem will be introduced as very related to the visual-textual grounding problem tackled in this Ph.D. thesis.

To solve the visual-grounding problem, usually, the models in the literature rely on continuous representations (i.e., features) of the textual phrases and bounding boxes. These features, which aim to convey information about the queries and the objects depicted in the bounding boxes, are used by the model to predict which bounding box is related to the query in input. Usually, the bounding boxes features are the output of an internal layer of the object detector, while the textual features are calculated from the sequence of words composing the query.

3.1 Supervised Visual Grounding

There is a vast literature about supervised visual-textual grounding. When considering the fusion of the textual features and bounding box features, multiple strategies are employed. Some approaches [74, 7] adopt methods such as Multi-layer Perceptron (MLP), while others adopt the cosine similarity for predicting alignment among bounding boxes and queries [75].

More complex strategies, such as Canonical Correlation Analysis (CCA) [76, 77], Multi-modal Compact Bilinear (MCB) [78], attention methodologies [79, 80, 6], and graph structures [81] are employed. H. Akbari *et al.* [80], instead of using bounding boxes to delimit objects in the image, they present a unique approach that predicts the location of image content referred by an input phrase through a heatmap, utilizing a multi-level multi-modal attention mechanism. Given a set of bounding box proposals and a textual sentence in input, A. Rohrbach *et al.* [6] proposes a model capable of selecting the optimal query-region pair through attention mechanics. The model assigns high attention to the bounding boxes that best match the query in the context of a given textual phrase. A pre-trained object detector is utilized for bounding box extraction.

Rather than emphasizing the fusion component, Z. Yu *et al.* [23] introduces a visual-textual grounding model with diverse and discriminative bounding box proposals that performs well without requiring a complex multi-modal fusion operator. In their work, they address the problem of the quality of the bounding box proposals which occurs when the two-stage approach is adopted. Indeed, if the bounding box proposals do not cover all the objects referred by the textual phrase, there is no hope that the grounding model will associate the correct regions of the images with the parts of the textual phrase. So, the core idea of Z. Yu *et al.* approach is to deploy an object detector able to recognize and discriminate many different objects depicted all over the image, even if the detection accuracy drops, and to use those proposals during grounding.

Instead, to overcome the problem of the quality of the bounding box proposals, Z. Yang *et al.* [24] designed a novel one-stage model. "if none of the candidates could cover the ground truth region", they argue, "there is no hope in the second stage to rank the right region to the top". So, they suggest a one-stage model that enables end-to-end joint optimization, focusing on integrating the text query's embedding into the YOLOv3 object detector, together with spatial features. Following a similar approach, A. Sadhu *et al.* [25] proposed a one-stage model focusing on the slightly different task of Zero-Shot Grounding, which includes unseen nouns in phrases. They claim that a two-stage approach is an obstacle due to the constrained generation of appropriate proposals. For this reason, they propose a one-stage model which combines the detector network and the grounding system to predict classification scores and bounding box regression parameters.

Lastly, H. Zhang *et al.* [8] adopts an entirely different methodology. Based on the variational Bayesian method, they capture context information by leveraging the reciprocal relation between the referent and its context.

3.2 Weakly-Supervised Visual Grounding

In weakly-supervised methods, some approaches cast the problem as a retrieval task [82, 83], while others adopt an encoder-decoder structure [84, 7, 85, 86, 6, 87], or employs a contrastive learning loss [9, 88, 89].

R. Hu *et al.* [82] proposed a model that returns bounding box proposals using an object retrieval approach where visual and textual information are integrated with contextual, spatial, and global visual features. While S. Datta *et al.* [83] proposed to learn to ground by optimizing the model for the downstream task of caption-to-image retrieval. The assumption is that to be able to solve the caption-to-image retrieval task, a model implicitly needs to perform visual-textual grounding. This new way to cast the visual-textual grounding problem allows the authors to train the model in a supervised setting for the caption-to-image retrieval task, and then deploy it to the visual-textual grounding task.

K. Chen *et al.* [7] tackled the weakly-supervised visual grounding problem by learning to reconstruct the input. In doing so, the model uses the visual information contained in proposals and the knowledge conveyed by the object detector. To attend to relevant features, they introduce the knowledge base pooling (KBP) component that aims to return a score between the query and the proposal's classification label. Similarly, S. A. Javed *et al.* [84] proposed a novel encoder-decoder framework for unsupervised visual-textual grounding, which uses concept learning to obtain self-supervision. The model initially selects all nouns from textual phrases through a POS tagger. These nouns are then used to train the model in an encoder-decoder style. The encoder localizes the region (as a heatmap) representing the concept in the batch, while the decoder reconstructs the concept through a classifier. F. Zhao *et al.* [85] proposed a spatial transformers [90]. Spatial transformers are convolutional neural networks that learn a new translation, scale, and rotation-invariant representation of features. Initially, the model reconstructs the input phrase while suppressing the reconstruction of different phrases for the same image, and then it predicts regions with similar spatial features. Instead, A. Rohrbach *et al.* extended their model [6] also to solve the visual-textual grounding task in a weakly-supervised setting. More in detail, their model extends their supervised model with a new neural network that aims to reconstruct the textual query from the features of the selected bounding box.

X. Liu *et al.* [86] proposed a reconstruction network based on an attention map. They first extract subject, location, and context features for both language and visual modalities. Language features are extracted by means of an attention mechanism applied to the sentence encoded with a Bi-LSTM network. Visual features are extracted from object detector convolutional layers, along with context features consisting of the position of the proposal relative to the image and neighbors' features. Through a reconstruction module, they learn to ground by optimizing the reconstruction with respect to sentences, attributes, subject, location, and context. Differently from previous works, A. Arbelle *et al.* [87] proposed a framework that solves the visual-textual grounding task by adopting the concept of alpha blending. Initially, given two images, the model composes a new image using pieces of both images and generates an alpha mask that keeps track of the composition. This mixed image is then fed into the encoder-decoder model, which also receives the two sentences associated with the images, and the objective is to reconstruct the alpha mask that separates the two images. At test time, the model interprets the input image as a composition of two images and produces a mask that separates the image regions concerning the two input queries, thus, localizing regions.

Regarding contrastive learning approaches, T. Gupta *et al.* [9] proposed a visual-textual grounding model that learns by contrastive examples. In particular, the model is trained to maximize the compatibility function among positive query and image pairs while minimizing other negative query and image pairs. Negative queries are built with a language model that substitutes noun words in the true textual phrase with contextually plausible but untrue words. Similarly, Q. Wang *et al.* [88] proposed a multimodal alignment framework (MAF) that joins the strength points of contrastive learning and uses object detector annotations. Inspired by [89], they employ the Faster-R CNN object detector trained on the Visual Genome dataset. Using a linear projection, they join proposal features with the word embedding of each proposal's classification label. The textual features are extracted with an attention mechanism that attends to the visual features. The final attention score is the maximum similarity obtained between the proposal and word features.

Other approaches embed the visual and textual features in the same embedding space [80, 91], sometimes enforcing preserving syntactic structures occurring in the textual phrase [92, 93, 83, 94]. H. Akbari *et al.* [80] addressed the problem of phrase localization by learning a multi-level joint semantic embedding space for both textual and visual modalities. The multi-level approach is implemented through an attention mechanism on top of multi-level visual features and contextualized text embeddings. Inspired by previous works on visual saliency that proved its effectiveness, V. Ramanishka *et al.* [91] applied the same approach to solve the visual-textual grounding task. Both the models of H. Akbari *et al.* and V. Ramanishka *et al.* output a heatmap. L. Wang *et al.* [92] proposed a new approach to generate joint embeddings for visual and textual modalities. However, in the embedding space, the authors enforce structure-preserving constraints which embed objects and images that share similar meanings near each other. S. Fidler *et al.* [93, 83] aimed at semantic scene understanding by incorporating both textual and visual information. Their model employs a natural language processing (NLP) parser to deconstruct textual phrases into nouns and prepositions, subsequently utilized to produce potentials in a holistic scene model. F. Xiao *et al.* [94] proposed a weakly-supervised approach that learns to visually ground phrases according to the linguistic structures of the textual phrase in input. More in detail, their model adopts an NLP parser to detect the parent-sibling and sibling-sibling linguistic constraints, which are then enforced in the visual modality.

Unlike the models presented before, J. Wang in *et al.* [89] proposes a method to solve the visual-textual grounding task without even performing any training on the visual-textual grounding model. Their core idea is to leverage several pre-trained object detectors to extract bounding box information and then compare that information with the query in input in a word embedding space. Objects detectors vary in number and type of categories.

While most of the approaches in the literature consider 2D images, C. Kont *et al.* [95] proposed a model that adopts RGB images with a depth channel (i.e., 3D). Their work, which adopts indoor images, aims to detect the class of the 3D objects and match nouns with the best referred visual objects. However, a limitation of their model is that it can classify only 21 different object classes.

Lastly, B. A. Plummer *et al.* [77] proposes a new approach, and at the same time, they release a new open-source dataset for visual-textual grounding, namely the Flickr30k Entities dataset. Their proposed approach is based on Canonical Correlation Analysis (CCA) and evaluates each region-phrase correspondence independently. In other words, their approach does not consider the joint reasoning among all region-phrase pairs' correspondences, which is often performed by most works adopting the weakly-supervised approach.

3.3 Visual-Textual-Knowledge Entity Linking (VTKEL) Problem

The Visual-Textual-Knowledge Entity Linking problem is an area of research very related to the visual-textual grounding problem. It is the most related research area to this Ph.D. thesis, which aims to use additional information to solve the visual-textual grounding.

In the following works [15, 16, 17], the authors introduce the problem of aligning the visual entities (i.e., the bounding boxes) and the textual entities (i.e., the noun of the textual phrase) with the nodes of the YAGO [96, 97, 98] knowledge graph. Always in these works, the authors proposed a baseline, namely VT-LINKER, to solve the problem and extended the Flickr30k Entities dataset with the ground truths alignment of queries and bounding boxes with knowledge graph nodes. Their model proposal's key idea is to move the alignment problem between bounding boxes and texts on the knowledge graph. More in detail, initially, their model deploys an NLP parser (PIKES [99]) to solve the named-entity recognition task on the textual phrase. The parser returns the node of the YAGO knowledge graph that most represent the entity expressed by the query. Then, the model adopts an object detector to locate and classify all the objects depicted in an image. Thanks to the bounding boxes classification labels and a predefined mapping function, the model retrieves the node of YAGO most related to the entity delimited by the bounding boxes. Finally, the alignment is performed on YAGO by observing whether parent-child relationships exist between the found nodes. If it exists then the corresponding visual and textual entities are also aligned.

4
The General Framework

This Ph.D. thesis aims to solve the visual-textual grounding task with additional prior information. While the classic method involves two input modalities (text and image), this thesis proposes incorporating a third modality in the form of a graph. The graph is a discrete structure that can represent any kind of information that can be used to solve the grounding task.

In the following, the visual-textual grounding task and a probabilistic framework designed to use additional knowledge will be formally defined . Then, a comprehensive analysis will be provided demonstrating that the proposed framework can effectively frame existing works in the literature as specific instances. In addition, it will be illustrated how the proposed framework can be employed to devise a novel approach to visual-textual grounding based on an innovative factorization of probabilities not yet explored in the literature.

4.1 Visual Grounding Formal Definition

Visual-textual grounding is the general task of locating the components of a structured description in an image. In order to solve this task, first, it is necessary to recognize all the objects in the image and the components in the text. After, it needs to find the correct alignment among the nouns and the objects.

Each object detected in the image is usually represented with a rectangle called bounding box, while each component detected in the text is usually called query. The bounding box is determined by its position in the image and by its dimension, while the query is determined by the position of the first character and the position of the last character in the input text.

Formally, given an image $\boldsymbol{I} \in \mathcal{I}$ and a sentence S $\in \mathcal{S}$ the visual-textual grounding task aims to learn a map $\gamma : \mathcal{I} \times \mathcal{S} \rightarrow 2^{\mathcal{Q}_\mathrm{S} \times \mathcal{B}_{\boldsymbol{I}}}$, where $\mathcal{Q}_\mathrm{S}$ is the domain of the noun phrases defined on S, and $\mathcal{B}_{\boldsymbol{I}}$ is the domain of all the bounding boxes defined on $\boldsymbol{I}$. So, given an image $\boldsymbol{I}$ containing e objects identified via the set of bounding boxes $B_{\boldsymbol{I}} = \{\boldsymbol{b}_i\}_{i=1}^{e}$, where $\boldsymbol{b}_i \in \mathbb{R}^4$ is the vector of coordinates identifying a bounding box in $\boldsymbol{I}$, and a sentence S containing m noun phrases gathered in the set $Q_\mathrm{S} = \{\boldsymbol{q}_j\}_{j=1}^{m}$, where $\boldsymbol{q}_j \in \mathbb{N}^2$ is a vector containing the initial and final character positions in the sentence S, $\gamma(\boldsymbol{I}, \mathrm{S})$ returns a subset $\Gamma \subseteq Q_\mathrm{S} \times B_{\boldsymbol{I}}$ where each couple $(\boldsymbol{q}, \boldsymbol{b}) \in \Gamma$ associates the noun phrase $\boldsymbol{q}$ to the bounding box $\boldsymbol{b}$. The object detector detects the bounding boxes $B_{\boldsymbol{I}}$ from a given image $\boldsymbol{I}$ while a natural language parser extracts the queries Q_S from a given sentence in S $\in \mathcal{S}$.

In the supervised setting, the model during training can utilize all the information available, which consists of a training set of n examples defined as $D = \{(\boldsymbol{I}_i, \mathrm{S}_i, \Gamma_i^{gt})\}_{i=1}^{n}$, where Γ_i^{gt} is the set of ground truth associations for the example i. In the weakly-supervised setting, the training set of n examples is defined as $\mathcal{D} = \{(\boldsymbol{I}_i, \mathrm{S}_i)\}_{i=1}^{n}$. In other words, during model training, only the information about sentence S_i describing the image $\boldsymbol{I}_i$ is available, while there is no information about which noun phrase $\boldsymbol{q} \in \mathcal{Q}_\mathrm{S}$ refers to each bounding box $\boldsymbol{b} \in \mathcal{B}_{\boldsymbol{I}}$ (i.e., Γ_i^{gt}).

This document aims to include also a knowledge graph in the resolution of the visual-textual grounding problem. The knowledge graph is defined as a directed graph $KG = (V, E, \Phi_l, \Phi_r, L, R)$ where:

- V: is the set of nodes, also called concepts in this document;
- E: is the set of direct edges between two nodes, also called relationships;
- Φ_l: is the function defined over edges that returns the start node of the edge:

$$\mathit{\Phi}_l : E \longrightarrow V$$

- Φ_r: is the function defined over edges that returns the end node of the edge:

$$\mathit{\Phi}_r : E \longrightarrow V$$

- L: is the function that associates to each node its label in Θ_v;

$$L : V \longrightarrow \Theta_v$$

- R: is the function that associates to each edge its label in Θ_e;

$$R : E \longrightarrow \Theta_e$$

- $\Theta_v \cap \Theta_e = 0$ i.e., there are no labels common to both sets.

Given a node $v \in V$, the following are defined:

$$\begin{aligned}
\mathcal{N}^+(v) &= \{u \mid u \in V, \exists\, e \in E, \Phi_r(e) = u \wedge \Phi_l(e) = v\}, \\
\mathcal{N}^-(v) &= \{u \mid u \in V, \exists\, e \in E, \Phi_r(e) = v \wedge \Phi_l(e) = u\}, \\
indegree &=\mid \mathcal{N}^+(v) \mid, \\
outdegree &=\mid \mathcal{N}^-(v) \mid .
\end{aligned}$$

The set of all neighborhoods of $v \in V$ is defined as:

$$\mathcal{N}(v) = \{u \mid u \in \mathcal{N}^+(v) \vee u \in \mathcal{N}^-(v)\}.$$

4.2 General Formulation

This section defines the new general framework intended to include graph information. More in detail, this section proposes a probabilistic framework that aims to learn the function $\gamma : \mathcal{I} \times \mathcal{S} \times KG \rightarrow 2^{\mathcal{Q}_\mathrm{S} \times \mathcal{B}_{\boldsymbol{I}} \times \mathcal{V}}$, where $\mathcal{Q}_\mathrm{S}$ is the domain of the noun phrases defined on S, $\mathcal{B}_{\boldsymbol{I}}$ is the domain of all the bounding boxes defined on $\boldsymbol{I}$, and $\mathcal{V}$ is the domain of the vertices V of the graph. In the rest of this chapter, when using the correct notation is not strictly necessary, B will denote $B_{\boldsymbol{I}}$, and Q will denote Q_S.

Let B be a set of bounding boxes of an image $\boldsymbol{I}$, Q a set of textual queries occurring in the image's caption, and V the set of vertices of a knowledge graph. The framework is interested in finding the alignments between the bounding boxes in B, the textual queries/mentions in Q, and the nodes in V. Intuitively, the alignment $\langle \boldsymbol{b}, \boldsymbol{q}, v \rangle \in B \times Q \times V$ represents the fact that the bounding box $\boldsymbol{b}$ shows an object mentioned by the query $\boldsymbol{q}$ and is (of type) v.

Since there is an amount of uncertainty in predicting these alignments, there is the need to develop a probabilistic framework that allows predicting the alignment $\langle \boldsymbol{b}, \boldsymbol{q}, v\rangle$ with an associated probability $\mathbb{P}\left(X_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}\right)$, where $X_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}$ is a boolean random variable taking values 0 if the $\langle \boldsymbol{b}, \boldsymbol{q}, v\rangle$ are not aligned and 1 if they are aligned.

To consider the joint distribution on all the alignments in $B \times Q \times V$, the set of boolean random variables is defined as $X_{B,Q,V} = \{X_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}\}_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle \in B\times Q\times V}$, and with $x_{B,Q,V} = \{x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}\}_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle \in B\times Q\times V}$ it is denoted an assignment to all the variables in $X_{B,Q,V}$. Notice that every value x of X corresponds to a subset of $B \times Q \times V$. When it is clear from the context, all the indexes are omitted. In general, the framework estimates:

$$\mathbb{P}\left(X = x\right). \tag{4.1}$$

Example 4.2.1 *Consider a picture showing two people one of which is walking a dog with the caption "John with Oscar: his wonderful golden retriever". Thus* $B = \{\boldsymbol{b}_{per1}, \boldsymbol{b}_{per2}, \boldsymbol{b}_{dog}\}$ *and* $Q = \{\boldsymbol{q}_{john}, \boldsymbol{q}_{oscar}, \boldsymbol{q}_{golden}\}$*. Furthermore, suppose that* V *contains the three nodes,* $V = \{v_{man}, v_{animal}, v_{dog}\}$*. In total, there are* 27 *triples, some of them are not very probable, and others, instead yes. There is an amount of* 2^{27} *possible alignments, where an alignment is a set of triples (not a single triple). In this case, the alignment with maximum probability should be:*

$$\left\{\langle \boldsymbol{b}_{per1}, \boldsymbol{q}_{john}, v_{man}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{oscar}, v_{dog}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{golden}, v_{dog}\rangle\right\}.$$

However, it might be uncertain if John is actually the other person, so another possible alignment is:

$$\left\{\langle \boldsymbol{b}_{per2}, \boldsymbol{q}_{john}, v_{man}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{oscar}, v_{dog}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{golden}, v_{dog}\rangle\right\},$$

but this alignment is less probable because the other person is in the background and far from the dog. However, it could be that the other person is indeed John himself reflecting in a window. Therefore, the mapping:

$$\left\{\langle \boldsymbol{b}_{per1}, \boldsymbol{q}_{john}, v_{man}\rangle \; \left\{\langle \boldsymbol{b}_{per2}, \boldsymbol{q}_{john}, v_{man}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{oscar}, v_{dog}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{golden}, v_{dog}\rangle\right\},\right.$$

has a non 0 *probability. Another uncertain concerns the fact that Oscar is the name of the dog* *or the other person, so another possible mapping is:*

$$\{\langle \boldsymbol{b}_{per1}, \boldsymbol{q}_{john}, v_{man}\rangle \; \langle \boldsymbol{b}_{per2}, \boldsymbol{q}_{oscar}, v_{man}\rangle \; \langle \boldsymbol{b}_{dog}, \boldsymbol{q}_{golden}, v_{dog}\rangle\}.$$

And so on ... it is possible to continue considering all the possible subsets of triples.

In the rest of the document, when it is not strictly necessary to use the correct notation, given some set of random variables X, the informal notation $\mathbb{P}(X)$ will be used to denote $\mathbb{P}(X = x)$, leaving implicit the assignment x.

Although reasoning with triples is very general, it turns out to be less intuitive and, it might be convenient to consider pairs of media at a time, instead of triplets. This has a price, i.e., some possible alignments are lost. Let's analyze the situation better.

4.2.1 The closure property

It is possible to consider the following restriction on the possible assignments of X.

Definition 1 (Closure) *An assignment $x_{\langle \mathbf{b},\boldsymbol{q},v\rangle}$ to $X_{\langle \mathbf{b},\boldsymbol{q},v\rangle}$ satisfies the closure condition if: $x_{\langle \boldsymbol{b},\boldsymbol{q},v'\rangle} = 1$, $x_{\langle \boldsymbol{b},\boldsymbol{q}',v\rangle} = 1$, and $x_{\langle \boldsymbol{b}',\boldsymbol{q},v\rangle} = 1$, implies that $x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = 1$, for every $\mathbf{b}, \boldsymbol{b}' \in B$, $\boldsymbol{q}, \boldsymbol{q}' \in Q$, and $v, v' \in V$. More compactly, this can be written with the fact that each assignment x should satisfy the following boolean formula:*

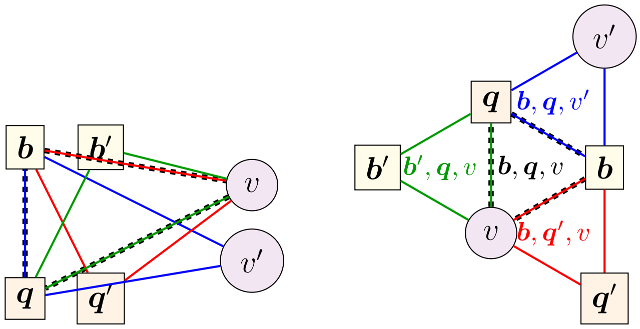
$$X_{\langle \boldsymbol{b},\boldsymbol{q},v'\rangle} \wedge X_{\langle \boldsymbol{b},\boldsymbol{q}',v\rangle} \wedge X_{\langle \boldsymbol{b}',\boldsymbol{q},v\rangle} \to X_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} \tag{4.2}$$

Graphically the closure condition is shown in Figure 4.1. Under the closure hypothesis, a more compact representation using three sets of random variables can be found:

$$\begin{aligned} X_{B,V} &= \{X_{\langle \boldsymbol{b},v\rangle}\}_{\langle \boldsymbol{b},v\rangle \in B\times V}, \\ X_{Q,V} &= \{X_{\langle \boldsymbol{q},v\rangle}\}_{\langle \boldsymbol{q},v\rangle \in Q\times V}, \\ X_{B,Q} &= \{X_{\langle \boldsymbol{b},\boldsymbol{q}\rangle}\}_{\langle \boldsymbol{b},\boldsymbol{q}\rangle \in B\times Q}, \end{aligned}$$

one for each pair of media (text, images, and knowledge graph). This is formally proved by the following proposition.

Proposition 4.2.1 *x satisfies* (4.2) *if and only if there are three assignments $x_{B,V}$, $x_{Q,V}$, and*



b
b'
v
q
q'
v'
v'
q
$\boldsymbol{b},\boldsymbol{q},v'$
b'
$\boldsymbol{b}',\boldsymbol{q},v$
$\boldsymbol{b},\boldsymbol{q},v$
b
v
$\boldsymbol{b},\boldsymbol{q}',v$
q'

$$\langle \boldsymbol{b},\boldsymbol{q},v'\rangle \wedge \langle \boldsymbol{b},\boldsymbol{q}',v\rangle \wedge \langle \boldsymbol{b}',\boldsymbol{q},v\rangle \rightarrow \langle \boldsymbol{b},\boldsymbol{q},v\rangle$$

Figure 4.1: The two graphs shown are equal. The second disposition better shows the property. The closure property states that the triangle obtained from the edges of three triangles is also a triangle. In the picture, the three triangles (blue, red and green) contribute to the construction of a fourth triangle with vertexes $\boldsymbol{b}$, $\boldsymbol{q}$ and v

$x_{B,Q}$ to $X_{B,V}$, $X_{Q,V}$ and $X_{B,Q}$ respectively, such that:

$$x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = x_{\langle \boldsymbol{b},v\rangle} \cdot x_{\langle \boldsymbol{q},v\rangle} \cdot x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle}, \tag{4.3}$$

for all $\langle \boldsymbol{b},\boldsymbol{q},v\rangle \in B \times Q \times V$.

Proof 4.2.1 *Suppose that x satisfies condition (4.2). Let's define:*

$$x_{\langle \boldsymbol{b},v\rangle} = \max_{\boldsymbol{q}} x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle},$$
$$x_{\langle \boldsymbol{q},v\rangle} = \max_{\boldsymbol{b}} x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle},$$
$$x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle} = \max_{v} x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}.$$

It will be proved that:

$$x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = x_{\langle \boldsymbol{b},v\rangle} \cdot x_{\langle \boldsymbol{q},v\rangle} \cdot x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle}. \tag{4.4}$$

Suppose that $x_{\langle \boldsymbol{b},v\rangle} = 0$ then for all $\boldsymbol{q} \in Q$, $x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = 0$. Similar arguments holds for $x_{\langle \boldsymbol{q},v\rangle}$ and $x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle}$. This implies that if one among the factors are 0 then $x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle}$ is equal to 0. Suppose that $x_{\langle \boldsymbol{b},v\rangle} = x_{\langle \boldsymbol{q},v\rangle} = x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle} = 1$ then there are $\mathbf{q}'$, $\mathbf{b}'$ and $\mathbf{v}'$ such that $x_{\langle \boldsymbol{b},\boldsymbol{q}',v\rangle} = x_{\langle \boldsymbol{b}',\boldsymbol{q},v\rangle} = x_{\langle \boldsymbol{b},\boldsymbol{q},v'\rangle} = 1$. From condition (4.2) it is obtained $x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = 1$.

Vice-versa. Suppose that for all $\langle \boldsymbol{b},\boldsymbol{q},v\rangle \in B \times Q \times V$, $x_{\langle \boldsymbol{b},\boldsymbol{q},v\rangle} = x_{\langle \boldsymbol{b},v\rangle} \cdot x_{\langle \boldsymbol{q},v\rangle} \cdot x_{\langle \boldsymbol{b},\boldsymbol{q}\rangle}$, *and let's prove condition* (4.2). *If* $x_{\langle \mathbf{b},\boldsymbol{q},v' \rangle} = x_{\langle \mathbf{b},\boldsymbol{q}',v \rangle} = x_{\langle \mathbf{b}',\boldsymbol{q},v \rangle} = 1$ *then, by construction of* x_{BV}*,* x_{QV} *and* x_{BQ} *it is obtained that* $x_{\langle \mathbf{b},v \rangle} = x_{\langle \boldsymbol{q},v \rangle} = x_{\langle \mathbf{b},\boldsymbol{q} \rangle} = 1$*, and therefore, by hypothesis* $x_{\langle \mathbf{b},\boldsymbol{q},v \rangle} = 1$.

Accepting the closure hypothesis, the entire probability distribution can be factorized on the joint distribution of the random boolean variables, one for each pair in $(B \times V) \cup (Q \times V) \cup (B \times Q)$. In other words, it is needed an estimation of:

$$\mathbb{P}\left(X_{B,V} = x_{B,V}, X_{Q,V} = x_{Q,V}, X_{B,Q} = x_{B,Q}\right), \tag{4.5}$$

that without any assumptions can be factored as follows:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}\right) = \mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}\right) \cdot \mathbb{P}\left(X_{B,V}, X_{Q,V}\right). \tag{4.6}$$

In the following section, it will be introduced some assumptions that will factorize the probability distribution further.

4.2.2 Additional Independent Assumptions

Estimating (4.6) is prohibitive, given the number of variables involved. The knowledge graph indeed can contain a huge number of nodes. If, for instance, WordNet is considered as a knowledge graph, then $|V| = 175,979$, which implies that a picture containing two bounding boxes with a caption with two textual mentions results in more than 700000 boolean variables. Therefore there is the need to consider a number of independent assumptions on the variables in X, that allow a factorization of (4.6).

4.2.2.1 Independence of $X_{B,V}$ from $X_{Q,V}$

A first independence assumption can be done by supposing that the alignment of the bounding boxes to the nodes of the knowledge graph, (i.e., the bounding box classification) is independent of the alignment of the textual concepts with the nodes of the knowledge graph (word sense disambiguation). This results in assuming that $\mathbb{P}\left(X_{B,V}, X_{Q,V}\right)$ factorizes in $\mathbb{P}\left(X_{B,V}\right) \cdot Pr(X_{Q,V})$. With this assumption, the following factorization of (4.6) is obtained:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}\right) = \mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}\right) \cdot \mathbb{P}\left(X_{B,V}\right) \cdot \mathbb{P}\left(X_{Q,V}\right). \tag{4.7}$$

This assumption removes the possibility to use the information of one media to interpret the other media. For instance, the possibility of using textual bias to help the classification of the bounding boxes in the image is lost. In fact, suppose that the text contains the relatively unambiguous word "dog". This constitutes a bias in the classification of the bounding boxes of the image, which could boost the label "dog" with respect to the other labels. When this independence of $X_{B,V}$ and $X_{Q,V}$ is considered, this boost is not possible.

4.2.2.2 Independence of $X_{\boldsymbol{b},V}$ from $X_{\boldsymbol{b}',V}$ (resp. of $X_{\boldsymbol{q},V}$ from $X_{\boldsymbol{q}',V}$)

A further independence hypothesis among the elements of $X_{B,V}$ (resp. $X_{Q,V}$) is based on the fact that the labeling of each bounding box (resp. textual mention) with the knowledge graph is independent of the alignments of the other bounding boxes (resp. textual mentions). This means that it is possible to assume:

$$\mathbb{P}\left(X_{B,V}\right) = \prod_{\boldsymbol{b}\in B} \mathbb{P}\left(X_{\boldsymbol{b},V}\right), \tag{4.8}$$

$$\mathbb{P}\left(X_{Q,V}\right) = \prod_{\boldsymbol{q}\in Q} \mathbb{P}\left(X_{\boldsymbol{q},V}\right), \tag{4.9}$$

where for every $\boldsymbol{b} \in B$, $X_{\boldsymbol{b},V}$ denotes the set of random variables $\{X_{\langle \boldsymbol{b},v\rangle}\}_{v\in V}$, and analogously for every $\boldsymbol{q} \in Q$, $X_{\boldsymbol{q},V}$ denotes the set of random variables $\{X_{\langle \boldsymbol{q},v\rangle}\}_{v\in V}$. Starting from the distribution 4.7, this new assumption results in the following factorization:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}\right) = \mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}\right) \cdot \prod_{\boldsymbol{b}\in B} \mathbb{P}\left(X_{\boldsymbol{b},V}\right) \cdot \prod_{\boldsymbol{q}\in Q} \mathbb{P}\left(X_{\boldsymbol{q},V}\right). \tag{4.10}$$

4.2.2.3 Conditional independence of $X_{B,Q}$

A further independence assumption can be obtained by assuming that the alignment of a bounding box $\boldsymbol{b}$ with the textual mention $\boldsymbol{q}$, depends only on the alignments of $\boldsymbol{b}$ and $\boldsymbol{q}$ with the knowledge graph, and not from the alignments of the other textual concepts and bounding boxes. This implies that the conditional probability $\mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}\right)$, can

Approach	Latent	Query	Bounding Boxes	Queries	Bounding Boxes
	Variables	Spatial Features	Spatial Features	Independencies	Independencies
B. A. Plummer *et al.* [77]	$X_{B,V}, X_{Q,V}$	✖	✖	✔	✔
Z. Yang *et al.* [24]	$X_{B,V}, X_{Q,V}$	✔	✖	✔	✖
DDPN [23]	$X_{B,V}, X_{Q,V}$	✖	✔	✔	✖
GroundeR [6]	$X_{B,V}, X_{Q,V}$	✖	✔	✔	✖
VT-LINKER [15]	None	✖	✖	✔	✔

Table 4.1: Summary of the differences in the framework instantiations outlined in Section 4.3. The "checkmark" and the "x" symbols refer to the presence or absence of the column item, respectively.

be factorized as follows:

$$\prod_{\substack{\boldsymbol{b}\in B\\ \boldsymbol{q}\in Q}} \mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid X_{\boldsymbol{b},V}, X_{\boldsymbol{q},V}\right). \tag{4.11}$$

With this further factorization the initial distribution (4.6) can be rewritten as:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}\right) = \prod_{\substack{\boldsymbol{b}\in B\\ \boldsymbol{q}\in Q}} \mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid X_{\boldsymbol{b},V}, X_{\boldsymbol{q},V}\right) \cdot \mathbb{P}\left(X_{\boldsymbol{b},V}\right) \cdot \mathbb{P}\left(X_{\boldsymbol{q},V}\right). \tag{4.12}$$

4.3 A Probabilistic Perspective of Some Models

This section presents some models in the literature under the probabilistic perspective given by the framework presented in Section 4.2. In particular, the presentation of each model focuses just on the factorization of the probability function, the visible and hidden random variables, the explicit and implicit independent assumptions, and how each function is estimated. Table 4.1 presents a synthetic view of the differences in the framework instantiations outlined in this section. Further detailed information on the approaches will not be discussed in this section, and reference should be made to the respective original sources.

The following terminology will be used during the presentation:

1. the symbol $\propto$ to represent the meaning of "proportional to";

2. the function $|\cdot|$ to represent the cardinality of a set;

3. the functions $||\cdot||_1$ and $||\cdot||_2$ to represent the absolute value norm and the Euclidean norm, respectively;

4. the notation $\overset{ind}{=}$ to indicate that equality under independence assumptions;

5. $Concat$ represents the concatenation function, while $Softmax$ refers to the softmax function.

For some random variable X, the informal notation $\mathbb{P}(X)$ will be used to denote the probability $\mathbb{P}(X = 1)$.

4.3.1 B. A. Plummer *et al.*

This section presents the model [77] that does not explicitly use the knowledge graph. Let $Z_B = \{\boldsymbol{Z_b}\}_{\boldsymbol{b} \in B}$ and $Z_Q = \{\boldsymbol{Z_q}\}_{\boldsymbol{q} \in Q}$ be two sets of observable continuous random vectors associated with the bounding boxes in B and the queries in Q. This model can be seen as an instantiation of the proposed framework in which $X_{Q,V}$ and $X_{B,V}$ are considered two latent variables. Starting from the probability function 4.5 and considering the two new variables, this approach estimates:

$$\begin{aligned}\int\!\!\int \mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V} \mid Z_B, Z_Q\right) \quad & dX_{B,V}\, dX_{Q,V}, \\ &= \mathbb{P}\left(X_{B,Q} \mid Z_B, Z_Q\right), \\ &\overset{ind}{=} \prod_{\substack{\boldsymbol{b} \in B \\ \boldsymbol{q} \in Q}} \mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid \boldsymbol{Z_b}, \boldsymbol{Z_q}\right),\end{aligned}$$

4.3.1.1 Estimating $\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid \boldsymbol{Z_b}, \boldsymbol{Z_q}\right)$

This function maps the bounding box features $\boldsymbol{Z_b}$ and the phrase features $\boldsymbol{Z_q}$ to a common space using the Canonical Correlation Analysis (CCA), where using the cosine distance function it predicts the distance between a pair of points. Let $(\boldsymbol{W_b}, \boldsymbol{W_q})$ a pair of matrices returned by the CCA, the probability function is estimated as:

$$\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} = 1 \mid \boldsymbol{Z_b} = \boldsymbol{z_b}, \boldsymbol{Z_q} = \boldsymbol{z_q}\right) \propto \left| \frac{\boldsymbol{z_b W_b} \cdot \boldsymbol{z_q W_q}}{\|\boldsymbol{z_b W_b}\|_2 \cdot \|\boldsymbol{z_q W_q}\|_2} \right|,$$

where $\boldsymbol{z_b}$ and $\boldsymbol{z_q}$ are the observable values of $\boldsymbol{Z_b}$ and $\boldsymbol{Z_q}$.

4.3.2 Z. Yang *et al.*

This section presents the model [24] which does not explicitly use the knowledge graph. This model can be seen as an instantiation of the proposed framework in which $X_{Q,V}$ and $X_{B,V}$ are considered two sets of latent variables. Let $Z_{\boldsymbol{I}} = \{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3\}$ be a set of three observable continuous random vectors associated with the image $\boldsymbol{I}$ in input at different resolution scales, $Z_Q = \{\boldsymbol{Z}_{\boldsymbol{q}}\}_{\boldsymbol{q} \in Q}$ be a set of observable continuous random vectors associated with the queries in Q and $\widehat{Z_Q} = \{\widehat{\boldsymbol{Z}_{\boldsymbol{q}}}\}_{\boldsymbol{q} \in Q}$ be a set of observable continuous random vectors representing the queries spatial features. Starting from the probability function 4.5 and considering the new variables, this approach estimates:

$$\begin{aligned} \int\int \mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V} \mid Z_{\boldsymbol{I}}, Z_Q, \widehat{Z_Q}\right) \quad & dX_{B,V}\, dX_{Q,V}, \\ &= \mathbb{P}\left(X_{B,Q} \mid Z_{\boldsymbol{I}}, Z_Q, \widehat{Z_Q}\right), \\ &\overset{ind}{=} \prod_{\boldsymbol{q} \in Q} \mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_{\boldsymbol{I}}, \boldsymbol{Z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}}\right). \end{aligned}$$

4.3.2.1 Estimating $\mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_{\boldsymbol{I}}, \boldsymbol{Z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}}\right)$

Let $x_{B,\boldsymbol{q}} = \{x_{\boldsymbol{b},\boldsymbol{q}}\}_{\boldsymbol{b} \in B}$ be the set of assignations where only one bounding box $\boldsymbol{b} \in B$ should grounded with the query $\boldsymbol{q} \in Q$, the probability distribution is estimated with a deep neural network NN:

$$\mathbb{P}\left(X_{B,\boldsymbol{q}} = x_{B,\boldsymbol{q}} \mid Z_{\boldsymbol{I}} = z_{\boldsymbol{I}}, \boldsymbol{Z}_{\boldsymbol{q}} = \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}} = \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right) \propto Softmax\left(NN\left(z_{\boldsymbol{I}}, \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right)\right),$$

where $z_{\boldsymbol{I}} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3\}$, $\boldsymbol{z}_{\boldsymbol{q}}$ and $\widehat{\boldsymbol{z}_{\boldsymbol{q}}}$ are the observable values of $Z_{\boldsymbol{I}}$, $\boldsymbol{Z}_{\boldsymbol{q}}$ and $\widehat{\boldsymbol{Z}_{\boldsymbol{q}}}$, respectively.

4.3.3 Diversified and Discriminative Proposal Networks model (DDPN)

This section presents the model [23] which does not explicitly use the knowledge graph. Let $Z_B = \{\boldsymbol{Z}_{\boldsymbol{b}}\}_{\boldsymbol{b} \in B}$ be a set of observable continuous random vectors associated with the bounding boxes in B, $Z_Q = \{\boldsymbol{Z}_{\boldsymbol{q}}\}_{\boldsymbol{q} \in Q}$ be a set of observable continuous random vectors associated with the queries in Q and $\widehat{Z_B} = \{\widehat{\boldsymbol{Z}_{\boldsymbol{b}}}\}_{\boldsymbol{b} \in B}$ be a set of observable continuous random vectors representing the bounding boxes spatial features. This model can be seen as an instantiation of the proposed framework in which $X_{Q,V}$ and $X_{B,V}$ are considered two latent variables. Starting from the probability function 4.5 and considering the new variables, this approach estimates:

$$
\begin{aligned}
\int\int \mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V} \mid Z_B, \widehat{Z_B}, Z_Q\right) \quad dX_{B,V}\, dX_{Q,V}, \\
= \mathbb{P}\left(X_{B,Q} \mid Z_B, \widehat{Z_B}, Z_Q\right), \\
\overset{ind}{=} \prod_{\boldsymbol{q}\in Q} \mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_B, \widehat{Z_B}, \boldsymbol{Z_q}\right).
\end{aligned}
$$

4.3.3.1 Estimating $\mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_B, \widehat{Z_B}, \boldsymbol{Z_q}\right)$

Let $x_{B,\boldsymbol{q}} = \{x_{\boldsymbol{b},\boldsymbol{q}}\}_{\boldsymbol{b}\in B}$ be the set of assignations where only one bounding box $\boldsymbol{b} \in B$ should grounded with the query $\boldsymbol{q} \in Q$, the probability distribution is estimated with a deep neural network NN:

$$
\begin{aligned}
\mathbb{P}\left(X_{B,\boldsymbol{q}} = x_{B,\boldsymbol{q}} \mid Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z_q} = \boldsymbol{z_q}\right) \\
\propto Softmax\left(NN\left(Concat(z_B, \widehat{z_B}, \boldsymbol{Z_q})\right)\right),
\end{aligned}
$$

where $z_B = \{\boldsymbol{z_b}\}_{\boldsymbol{b}\in B}$, $\widehat{z_B} = \{\widehat{\boldsymbol{z_b}}\}_{\boldsymbol{b}\in B}$, and $\boldsymbol{z_q}$ are the observable values of Z_B, $\widehat{Z_B}$ and $\boldsymbol{Z_q}$, respectively. This implies:

$$
\begin{aligned}
\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} = 1 \mid Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z_q} = \boldsymbol{z_q}\right) \\
\propto \frac{\exp\left(NN\left(Concat(\boldsymbol{z_b}, \widehat{\boldsymbol{z_b}}, \boldsymbol{z_q})\right)\right)}{\sum_{\boldsymbol{i}\in B} \exp\left(NN\left(Concat(\boldsymbol{z_i}, \widehat{\boldsymbol{z_i}}, \boldsymbol{z_q})\right)\right)}.
\end{aligned}
$$

4.3.4 GrounderR

This section presents the model [6] which does not explicitly use the knowledge graph. Let $Z_B = \{\boldsymbol{Z_b}\}_{\boldsymbol{b}\in B}$ be a set of observable continuous random vectors associated with the bounding boxes in B, $Z_Q = \{\boldsymbol{Z_q}\}_{\boldsymbol{q}\in Q}$ be a set of observable continuous random vectors associated with the queries in Q and $\widehat{Z_B} = \{\widehat{\boldsymbol{Z_b}}\}_{\boldsymbol{b}\in B}$ be a set of observable continuous random vectors representing the bounding boxes spatial features. This model can be seen as an instantiation of the proposed framework in which $X_{Q,V}$ and $X_{B,V}$ are considered two latent variables. Starting from the probability function 4.5 and considering the new variables, this approach estimates:

$$\int \int \mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V} \mid Z_B, \widehat{Z_B}, Z_Q\right) \quad dX_{B,V}\, dX_{Q,V},$$
$$= \mathbb{P}\left(X_{B,Q} \mid Z_B, \widehat{Z_B}, Z_Q\right),$$
$$\overset{ind}{=} \prod_{\boldsymbol{q}\in Q} \mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_B, \widehat{Z_B}, \boldsymbol{Z_q}\right).$$

4.3.4.1 Estimating $\mathbb{P}\left(X_{B,\boldsymbol{q}} \mid Z_B, \widehat{Z_B}, \boldsymbol{Z_q}\right)$

Let $x_{B,\boldsymbol{q}} = \{x_{\boldsymbol{b},\boldsymbol{q}}\}_{\boldsymbol{b}\in B}$ be the set of assignations where only one bounding box $\boldsymbol{b} \in B$ should grounded with the query $\boldsymbol{q} \in Q$, the probability distribution is estimated as:

$$\mathbb{P}\left(X_{B,\boldsymbol{q}} = x_{B,\boldsymbol{q}} \mid Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z_q} = \boldsymbol{z_q}\right)$$
$$\propto Softmax\left(NN\left(Concat\left(z_B, \widehat{z_B}\right), \boldsymbol{z_q}\right)\right),$$

where $z_B = \{\boldsymbol{z_b}\}_{\boldsymbol{b}\in B}$, $\widehat{z_B} = \{\widehat{\boldsymbol{z_b}}\}_{\boldsymbol{b}\in B}$, and $\boldsymbol{z_q}$ are the observable values of Z_B, $\widehat{Z_B}$ and $\boldsymbol{Z_q}$, respectively. This implies that:

$$\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} = 1 \mid Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z_q} = \boldsymbol{z_q}\right)$$
$$\propto \frac{\exp\left(NN\left(Concat(\boldsymbol{z_b}, \widehat{\boldsymbol{z_b}}), \boldsymbol{z_q}\right)\right)}{\sum_{\boldsymbol{i}\in B} \exp\left(NN\left(Concat(\boldsymbol{z_i}, \widehat{\boldsymbol{z_i}}), \boldsymbol{z_q}\right)\right)},$$

4.3.5 VT-LINKER

This section presents the existing baseline VT-LINKER presented in [15], in which the alignment between a query $\boldsymbol{q} \in Q$ and a bounding box $\boldsymbol{b} \in B$ is done with an algorithm. Starting from equation 4.12, this model estimates:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}\right) = \prod_{\substack{\boldsymbol{b}\in B\\ \boldsymbol{q}\in Q}} \mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid X_{\boldsymbol{b},V}, X_{\boldsymbol{q},V}\right) \cdot \mathbb{P}\left(X_{\boldsymbol{b},V}\right) \cdot \mathbb{P}\left(X_{\boldsymbol{q},V}\right)$$

4.3.5.1 Estimating $\mathbb{P}\left(X_{\boldsymbol{b},V}\right)$

This probability function is estimated using an object classifier in which, given a bounding box $\boldsymbol{b} \in B$, its predicted class corresponds to a unique node $v_{\boldsymbol{b}}$ in the knowledge graph. Then, for each bounding box, there is only a possible assignment $x^{'}_{\boldsymbol{b},V}$ to the knowledge graph nodes that returns probability 1:

$$\mathbb{P}\left(X_{\boldsymbol{b},V} = x_{\boldsymbol{b},V}\right) = \begin{cases} 1, & \text{if } x_{\boldsymbol{b},V} = x^{'}_{\boldsymbol{b},V}; \\ 0, & \text{otherwise.} \end{cases}$$

4.3.5.2 Estimating $\mathbb{P}\left(X_{\boldsymbol{q},V}\right)$

This probability function is estimated using a named-entity recognition system which returns, for each query $\boldsymbol{q} \in Q$, the unique knowledge graph node $v_{\boldsymbol{q}}$ that represents its information. Then, similarly to the bounding box case, for each query, there is only a possible assignment $x^{'}_{\boldsymbol{q},V}$ to the knowledge graph nodes that returns probability 1:

$$\mathbb{P}\left(X_{\boldsymbol{q},V} = x_{\boldsymbol{q},V}\right) = \begin{cases} 1, & \text{if } x_{\boldsymbol{q},V} = x^{'}_{\boldsymbol{q},V}; \\ 0, & \text{otherwise.} \end{cases}$$

4.3.5.3 Estimating $\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} \mid X_{\boldsymbol{b},V}, X_{\boldsymbol{q},V}\right)$

This function is implemented with a fixed algorithm that checks if the type of the node representing the query $\boldsymbol{q} \in Q$, and the type of the node representing the bounding box $\boldsymbol{b} \in B$, are related to each other with a relation of sub-class. Let $C\left(v\right)$ be the function that returns the type of a node $v \in V$ in the knowledge graph and $\sqsubseteq$ the symbol indicating the sub-class relation:

$$\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} = 1 \mid X_{\boldsymbol{b},V}, X_{\boldsymbol{q},V}\right) = \begin{cases} 1, & \text{if } C\left(v_{\boldsymbol{b}}\right) \sqsubseteq C\left(v_{\boldsymbol{q}}\right) \text{ or } C\left(v_{\boldsymbol{q}}\right) \sqsubseteq C\left(v_{\boldsymbol{b}}\right); \\ 0, & \text{otherwise.} \end{cases}$$

4.4 An Innovative Probability Distribution Factorization

As seen from the previous Section 4.3, generally, the approaches in literature treat the knowledge graph information as latent variables. Therefore, often models solve the visual-textual grounding by adopting a similar probability distribution, where only the method used to estimate the distributions tends to change. Based on this observation, this section presents a new model proposal to solve the visual-textual grounding task that adopts the probabilistic framework presented in Section 4.2.

Let $Z_B = \{\boldsymbol{Z_b}\}_{\boldsymbol{b} \in B}$ and $\widehat{Z_B} = \{\widehat{\boldsymbol{Z_b}}\}_{\boldsymbol{b} \in B}$ be two sets of observable continuous random vectors associated with the bounding boxes in B. In particular, Z_B is the set of bounding boxes features, while $\widehat{Z_B}$ is the set of the bounding boxes spatial features. Let $Z_Q = \{\boldsymbol{Z_q}\}_{\boldsymbol{q} \in Q}$ and $\widehat{Z_Q} = \{\widehat{\boldsymbol{Z_q}}\}_{\boldsymbol{q} \in Q}$ be two sets of observable continuous random vectors associated with the queries in Q. In particular, Z_Q is the set of queries features, while $\widehat{Z_Q}$ is the set of the queries spatial features. Considering the new variables and the knowledge graph KG, the following distribution should be estimated:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}, \mid Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}, KG\right),$$

which can be factorized as:

$$\mathbb{P}\left(X_{B,Q}, X_{B,V}, X_{Q,V}, \mid Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}, KG\right) = \Psi_1 \cdot \Psi_2 \cdot \Psi_3,$$

where:

$$\begin{aligned}
\Psi_1 &= \mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}, Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}, KG\right), \\
\Psi_2 &= \mathbb{P}\left(X_{B,V} \mid X_{Q,V}, Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}, KG\right), \\
\Psi_3 &= \mathbb{P}\left(X_{Q,V} \mid Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}, KG\right).
\end{aligned}$$

At this point, regarding the factor Ψ_1 it can be supposed that $X_{B,Q}$ and KG are conditionally independent given the variables $X_{B,V}$ and , $X_{Q,V}$, which is reasonable due to the fact that the grounding should focus more on the concepts referenced by the bounding boxes and queries instead of all the KG:

$$\Psi_1 = \mathbb{P}\left(X_{B,Q} \mid X_{B,V}, X_{Q,V}, Z_B, \widehat{Z_B}, Z_Q, \widehat{Z_Q}\right).$$

In addition, regarding the factors Ψ_2 and Ψ_3, it can be assumed that $X_{Q,V}$ is independent from the bounding boxes Z_B and $\widehat{Z_B}$ and that $X_{B,V}$ is independent from Z_Q and $\widehat{Z_Q}$:

$$\Psi_2 = \mathbb{P}\left(X_{B,V} \mid X_{Q,V}, Z_B, \widehat{Z_B}, KG\right),$$
$$\Psi_3 = \mathbb{P}\left(X_{Q,V} \mid Z_Q, \widehat{Z_Q}, KG\right),$$

and that the bounding boxes are independent of each other in factor Ψ_2:

$$\Psi_2 = \prod_{\boldsymbol{b} \in B} \mathbb{P}\left(X_{\boldsymbol{b},V} \mid X_{Q,V}, \boldsymbol{Z_b}, \widehat{\boldsymbol{Z_b}}, KG\right).$$

To conclude, supposing that the queries are independent of each other, the following probability distribution factorization should be estimated:

$$\prod_{\boldsymbol{q} \in Q} \mathbb{P}\left(X_{B,\boldsymbol{q}} \mid X_{B,V}, X_{\boldsymbol{q},V}, Z_B, \widehat{Z_B}, \boldsymbol{Z_q}, \widehat{\boldsymbol{Z_q}}\right) \cdot \prod_{\boldsymbol{b} \in B} \mathbb{P}\left(X_{\boldsymbol{b},V} \mid X_{\boldsymbol{q},V}, \boldsymbol{Z_b}, \widehat{\boldsymbol{Z_b}}, KG\right)$$
$$\cdot\, \mathbb{P}\left(X_{\boldsymbol{q},V} \mid \boldsymbol{Z_q}, \widehat{\boldsymbol{Z_q}}, KG\right).$$

This new probability distribution factorization leads to a two-stage visual-textual grounding approach, where estimating each component of the distribution is not straightforward. In the following, an estimation for each component will be proposed.

4.4.0.1 Estimating $\mathbb{P}\left(X_{\boldsymbol{q},V} \mid \boldsymbol{Z_q}, \widehat{\boldsymbol{Z_q}}, KG\right)$

This probability function is estimated using a word sense disambiguator system which returns, for each query $\boldsymbol{q} \in Q$, the unique knowledge graph node $v_{\boldsymbol{q}}$ that represents its information. Then, for each query, there is only a possible assignment $x^{'}_{\boldsymbol{q},V}$ to the knowledge graph nodes that returns probability 1:

$$\mathbb{P}\left(X_{\boldsymbol{q},V} = x_{\boldsymbol{q},V} \mid \boldsymbol{Z_q}, \widehat{\boldsymbol{Z_q}}, KG\right) = \begin{cases} 1, & \text{if } x_{\boldsymbol{q},V} = x^{'}_{\boldsymbol{q},V}; \\ 0, & \text{otherwise.} \end{cases}$$

In the literature, there are several word sense disambiguator systems [100, 101], such as EWISE [102] and EWISER [103].

4.4.0.2 Estimating $\mathbb{P}\left(X_{B,\boldsymbol{q}} \mid X_{B,V}, X_{\boldsymbol{q},V}, Z_B, \widehat{Z_B}, \boldsymbol{Z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}}\right)$

Let $x_{B,\boldsymbol{q}} = \{x_{\boldsymbol{b},\boldsymbol{q}}\}_{\boldsymbol{b} \in B}$ be the set of assignations where only one bounding box $\boldsymbol{b} \in B$ should grounded with the query $\boldsymbol{q} \in Q$, the probability distribution can be estimated using a neural network NN as:

$$\mathbb{P}\left(X_{B,\boldsymbol{q}} = x_{B,\boldsymbol{q}} \mid X_{B,V} = x_{B,V}, X_{\boldsymbol{q},V} = x_{\boldsymbol{q},V}, Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z}_{\boldsymbol{q}} = \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}} = \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right) \\ \propto Softmax\left(NN\left(x_{B,V}, x_{\boldsymbol{q},V}, z_B, \widehat{z_B}, \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right)\right),$$

where $z_B = \{\boldsymbol{z}_{\boldsymbol{b}}\}_{\boldsymbol{b} \in B}$, $\widehat{z_B} = \{\widehat{\boldsymbol{z}_{\boldsymbol{b}}}\}_{\boldsymbol{b} \in B}$, $\boldsymbol{z}_{\boldsymbol{q}}$, and $\widehat{\boldsymbol{z}_{\boldsymbol{q}}}$ are the observable values of Z_B, $\widehat{Z_B}$, $\boldsymbol{Z}_{\boldsymbol{q}}$, and $\widehat{\boldsymbol{Z}_{\boldsymbol{q}}}$, respectively. This implies that:

$$\mathbb{P}\left(X_{\boldsymbol{b},\boldsymbol{q}} = x_{\boldsymbol{b},\boldsymbol{q}} \mid X_{B,V} = x_{B,V}, X_{\boldsymbol{q},V} = x_{\boldsymbol{q},V}, Z_B = z_B, \widehat{Z_B} = \widehat{z_B}, \boldsymbol{Z}_{\boldsymbol{q}} = \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{Z}_{\boldsymbol{q}}} = \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right) \\ \propto \frac{\exp\left(NN\left(x_{\boldsymbol{b},V}, x_{\boldsymbol{q},V}, \boldsymbol{z}_{\boldsymbol{b}}, \widehat{\boldsymbol{z}_{\boldsymbol{b}}}, \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right)\right)}{\sum_{\boldsymbol{i} \in B} \exp\left(NN\left(x_{\boldsymbol{i},V}, x_{\boldsymbol{q},V}, \boldsymbol{z}_{\boldsymbol{i}}, \widehat{\boldsymbol{z}_{\boldsymbol{i}}}, \boldsymbol{z}_{\boldsymbol{q}}, \widehat{\boldsymbol{z}_{\boldsymbol{q}}}\right)\right)},$$

This estimation resembles those presented in Section 4.3.3 and Section 4.3.4 with the only exception of having more observable variables conditioning the resolution of the visual-textual grounding task. Future works will extend approaches in the literature to include this additional observable information.

4.4.0.3 Estimating $\mathbb{P}\left(X_{\boldsymbol{b},V} \mid X_{\boldsymbol{q},V}, \boldsymbol{Z}_{\boldsymbol{b}}, \widehat{\boldsymbol{Z}_{\boldsymbol{b}}}, KG\right)$

This probability function can be estimated using an object detector that: (i) locates and classifies the objects in the image conditioned by KG and the variables $X_{\boldsymbol{q},V}$; and (ii) align the bounding boxes to the knowledge graph nodes (i.e., $X_{\boldsymbol{b},V}$) according to their predicted classes, as done in VT-LINKER (Section 4.3.5). Thus, when the bounding boxes are located and classified, each predicted class corresponds to a unique node $v_{\boldsymbol{b}}$ in the knowledge graph, and for each bounding box, there is only a possible assignment $x^{'}_{\boldsymbol{b},V}$ to the knowledge graph nodes that returns probability 1:

$$\mathbb{P}\left(X_{\boldsymbol{b},V} = x_{\boldsymbol{b},V} \mid X_{\boldsymbol{q},V} = x_{\boldsymbol{q},V}, \boldsymbol{Z}_{\boldsymbol{b}} = \boldsymbol{z}_{\boldsymbol{b}}, \widehat{\boldsymbol{Z}_{\boldsymbol{b}}} = \widehat{\boldsymbol{z}_{\boldsymbol{b}}}, KG\right) = \begin{cases} 1, & \text{if } x_{\boldsymbol{b},V} = x^{'}_{\boldsymbol{b},V}; \\ 0, & \text{otherwise.} \end{cases}$$

It is evident that the classes of the object detector are very important in determining the alignment with the nodes of the graph. Object detectors should find all objects in the image and classify them correctly. However, the most common approach in the State-of-the-Art is to use the Bottom-Up [66] object detector, which is a model trained to identify 1600 different classes. These classes are the result of an automatic process that introduced some noise on the set of classes that may result in a sub-optimal representational space and likely impair the ability of the model to classify objects correctly. For this reason, Section 7.1 proposes a new slim set of less noisy classes that allow for a better estimate of the class probabilities of the bounding boxes.

A thorough search of the relevant literature yielded that an object detector that uses the information of a graph KG and the variables $x_{\boldsymbol{q},V}$ to detect and classify the objects in the images is still to be explored. For this reason, Section 7.2 proposes a method that can exploit that information, and that can be used to estimate this last probability function.

4.5 Summary of the Assumptions Made in Modeling

The probabilistic framework presented in this chapter aims to learn the probability distribution that models the alignment among triples made of images' regions, textual phrases, and a knowledge graph's nodes. Without any assumption, the estimation of this distribution is prohibitive given the high number of variables involved. For this reason, several assumptions were introduced in this chapter to make its estimation feasible, although some generalization of the approach has been sacrificed. The assumptions made in modeling the proposed probability distribution factorization are summarized below.

The first assumption regards the acceptance of the Closure 4.2 hypothesis, which makes the framework more intuitive and tractable, albeit sacrificing some possible alignments. The Closure hypothesis states that the triangle obtained from the edges of three triangles is also a triangle, which may not be true in the general case.

The second assumption states that the alignment among queries and bounding boxes are conditionally independent from the knowledge graph given the alignment of the queries and bounding boxes with the knowledge graph. This is reasonable due to the fact that the grounding should focus more on the boxes and queries instead of the knowledge graph.

The third assumption regards the independence of the alignment of the textual concepts with the knowledge graph from the alignment of the bounding boxes with the knowledge graph. This assumption removes the possibility to use the image information to interpret the textual information, although in some cases, visual information can be useful to disambiguate words' meanings.

To conclude, the last assumptions suppose that (i) the queries are always independent of each other and that (i) the alignment of each bounding box with the knowledge graph is independent of the others. The former assumption is reasonable in the visual-textual grounding area of research, where the queries are usually independent of the others during evaluation. Instead, the latter assumption removes the possibility of reasoning about all the objects appearing in the image when finding their alignment with the knowledge graph.

5
A Better Loss for Visual-Textual Grounding

As presented in the introduction of this Ph.D. thesis, while developing the probabilistic framework, the traditional visual-textual grounding task, which considers only two modalities (i.e., image and text), was also studied. In particular, the first contribution in this direction is the proposal of a new training loss for training deep learning models in the supervised setting.

In the last years, several works have addressed the visual-textual grounding problem by proposing more and more large and complex models that try to capture visual-textual dependencies better than before. These models are typically constituted by two main components that focus on how to learn useful multi-modal features for grounding and how to improve the predicted bounding box of the visual mention, respectively. Finding the right learning balance between these two sub-tasks is not easy, and the current models are not necessarily optimal with respect to this issue.

More in detail, this chapter[1] proposes a loss function based on bounding boxes classes probabilities that: (i) improves the bounding boxes selection; (ii) improves the bounding boxes coordinates prediction. The proposed model, although using a simple multi-modal feature fusion component, is able to achieve higher accuracy than State-of-the-Art models on two widely adopted datasets, reaching a better learning balance between the two sub-tasks mentioned above.

[1] Part of this work is published in [10].

5.1 Introduction

The visual-textual grounding problem is defined as the task of locating the content of the image referenced by a given sentence and it is a building block for many real-world applications and complex tasks. It is a challenging task, which requires a semantic understanding of the image content and its textual description, requiring the ability to predict the parts of the image content referred by a specific descriptive sentence. It can be formulated as an object detection task followed by a classification task in which, given an input image and sentence, the goal is to return only the detected object(s) in the image that represent(s) the best semantic match with the sentence. In the initial phase of research on this problem, many works have followed this formulation, developing the so-called two-stage approach models [6, 23], while more recent works have chosen to address the problem by a one-stage approach model, in which the object detection and the classification problem are solved jointly [24, 25].

In the literature, there are many works adopting increasingly improved object proposals and increasingly complex architectures than before in order to capture visual and textual information. These models are typically constituted by two main components that focus on how to learn useful multi-modal features for grounding and how to improve the predicted bounding box of the visual mention, respectively. Finding the right learning balance between these two sub-tasks is not easy, and the current models are not necessarily optimal with respect to this issue. This chapter proposes a model that, although using a simple multi-modal feature fusion component, is able to reach a higher accuracy than State-of-the-Art models thanks to the adoption of a more effective loss function that reaches a better learning balance between the two sub-tasks mentioned above.

The main contributions can be summarized as follows: (i) the proposal of a new loss for visual bounding box proposals, which also considers the object proposals' semantic information, differently from the works in the literature that just consider their shapes and spatial positions in the image; (ii) the proposal of a new regression loss on the bounding boxes coordinates, which is applied to a subset of all the proposals selected by considering the object proposals' semantic information. This loss differs from the one used by the approaches in the literature, which only considers the proposal with the largest overlap with the ground truth; (iii) this is the first approach that adopt the *Complete Intersection over Union* [104] loss for the visual-textual grounding task; (iv) it is experimentally shown that the proposed losses improve the performance of State-of-the-Art models.

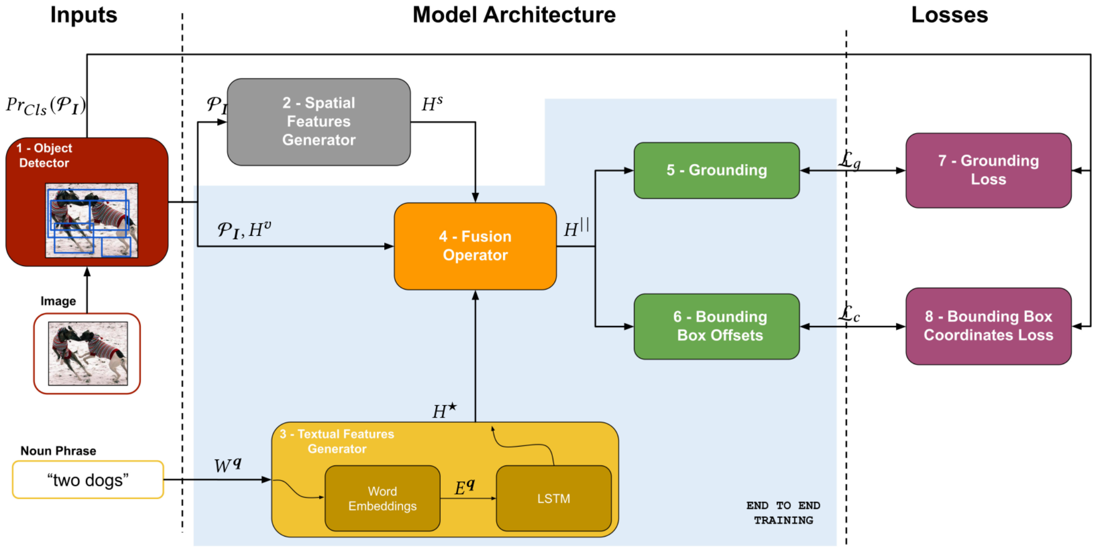


Figure 5.1: The two-stage model architecture overview. (**1**) Initially, the image is processed by a pre-trained *Faster R-CNN* object detector in order to extract all the proposals bounding boxes from which (**2**) the spatial features are generated. Then, the model (**3**) generates the textual features from the input noun phrase using the *Textual Features Generator* module by first retrieving each word embedding and then using an LSTM network. Finally, the model (**4**) fuses together all the visual, spatial, and textual features by the *Fusion Operator*, obtaining new features that are then used in the (**5**) *Grounding* and (**6**) *Bounding Box Offsets* modules, respectively. The defined losses $\mathcal{L}_g$ (**7**) and $\mathcal{L}_c$ (**8**) are used in order to train the network end-to-end on the components included in the light blue background.

5.2 Problem Definition

This chapter tackles the supervised visual-textual grounding task whose formal definition is introduced in Section 4.1. Please, notice that the same noun phrase can be associated with several different bounding boxes, as well as the same bounding box can be associated with many different noun phrases. Following the current literature, in this chapter, it is assumed that each noun phrase is associated with one and only one bounding box. A bounding box, however, can identify more objects, e.g., several persons, in the case the noun phrase is "people".

Bear in mind that for model training, all the training set annotations can be used, which consist of a set of n examples defined as $D = \{(\boldsymbol{I}_i, \mathrm{S}_i, \Gamma_i^{gt})\}_{i=1}^{n}$, where Γ_i^{gt} is the set of ground truth associations, for example, i.

5.3 The Model Proposal

This section will first describe the model structure and then the training procedure that exploits the original part of this proposal, e.g., a loss function composed of novel sub-losses.

5.3.0.1 Model

The model proposal, outlined in Figure 5.1, follows a typical basic architecture for visual-textual grounding tasks. It is based on a two-stage approach in which, initially, a pre-trained object detector is used to extract, from a given image $\boldsymbol{I}$, a set of e bounding box proposals $\mathcal{P}_{\boldsymbol{I}}$, jointly with visual features H^v. The features represent the internal object detector activation values before the classification and regression layers for bounding boxes. Moreover, the model extracts the spatial features H^s from the bounding box proposals. It is also assumed that the object detector returns, for each bounding box proposal $\boldsymbol{p}_i \in \mathcal{P}_{\boldsymbol{I}}$, a probability distribution $Pr_{Cls}(\boldsymbol{p}_i)$ over a set Cls of predefined classes, i.e., the probability for each class $\xi \in Cls$ that the content of the bounding box proposal $\boldsymbol{p}_i$ belongs to ξ. This information is typically returned by most of the object detectors, and it will be used to define the novel loss terms.

Regarding the textual features extraction, given a noun phrase $\boldsymbol{q}_j$, all its words $W^{\boldsymbol{q}_j}$ are initially embedded in a set of vectors $E^{\boldsymbol{q}_j}$. Then, the model applies a LSTM [105] neural network to generate only one new embedding $\boldsymbol{h}^{\star}_j$ from the sequence of word embeddings for each phrase $\boldsymbol{q}_j$. Once vector $\boldsymbol{h}^{\star}_j$ has been generated from the noun phrase $\boldsymbol{q}_j$, the model performs a multi-modal feature fusion operation in order to combine the information contained in $\boldsymbol{h}^{\star}_j$ with each of the bounding box proposals $\boldsymbol{h}^v_z \in H^v$. This operation is implemented with a simple function that merges the multi-modal features together rather than relying on a more complex operator, such as bilinear-pooling [78] or deeper neural network architectures. Future works will use a more complex fusion operator that will lead to further improvements. The multi-modal fusion component returns the set of new vectorial representations $H^{||}$.

Finally, the model predicts the probability $\boldsymbol{P}_{jz}$ that a given noun phrase $\boldsymbol{q}_j$ is referred to the bounding box proposal $\boldsymbol{p}_z$. Indeed, the representations of the bounding box proposal's features conditioned with the textual features can also be used to refine the bounding box proposal's coordinates generated by the object detector independently of the textual features. Specifically, the model does not predict new bounding box coordinates but offsets the coordinates.

Technical details regarding the model are reported in Appendix B.1.

5.3.0.2 Training

This section presents the main novel contribution of this chapter, i.e., a loss function composed of novel terms. The basic idea is to exploit the semantic information associated with bounding box proposals, i.e., the probability distribution over classes of the content of a bounding box returned by the object detector, in both the loss term concerning the grounding and the loss term concerning the refinement of the bounding box coordinates. In fact, differently from most of the previous works that use the *cross-entropy (CE)* loss or the standard *Kullback–Leibler(KL) divergence* loss for grounding, the model proposed in this chapter implements a KL divergence loss in which the ground truth probability is built also considering $Pr_{Cls}(\boldsymbol{p}_i)$ with $\boldsymbol{p}_i \in \mathcal{P}_{\boldsymbol{I}}$. Moreover, regarding the bounding boxes coordinates refinement, differently from previous works that use the *Smooth*$_{L1}$ *loss*, the model presented in this chapter adopts the *Complete Intersection over Union (CIoU) loss* [104]. See Appendix A.3 for more details about the *Intersection over Union (IoU)* metric and Appendix A.4 for more details about the *CIoU* metric. This is the first work adopting the *CIoU* loss to refine the final bounding box coordinates. Another difference with respect to all the refinement losses available in the literature is that the proposed loss does not restrict the coordinates refinement only to the best proposal coordinates, but extends the refinement to the subset of proposals that significantly overlap (according to a hyper-parameter) the ground truth, modulating the refinement by the agreement between the class probability of the best proposal and the class probability of the considered proposal. For the sake of presentation, the new loss terms are defined in the following, referring to a single example. The total loss is then obtained by summing up the contributions of all examples in the training set.

Given a training example $(\boldsymbol{I}, \mathrm{S}, \Gamma^{gt})$, and the bounding box proposals set $\mathcal{P}_{\boldsymbol{I}}$, the proposed loss function $\mathcal{L}$ (for a single example) is defined as:

$$\mathcal{L} = \mathcal{L}_g(\boldsymbol{P}, \mathcal{P}_{\boldsymbol{I}}, \Gamma^{gt}) + \lambda \mathcal{L}_c(\mathcal{P}_{\boldsymbol{I}}, \Gamma^{gt}),$$

where $\mathcal{L}_g$ is the loss used to “shape” the grounding distribution of proposals for each specific query in input, i.e., the probability that a given proposal is associated with a given query, $\mathcal{L}_c$ is the loss related to the refinement of the bounding boxes coordinates, and λ is a trade-off parameter.

Specifically, given m the number of noun phrases and e the number of bounding box proposals, the entries $(j \in [1, \dots, m], z \in [1, \dots, e])$ of matrix $\boldsymbol{U}$ are defined as:

$$\boldsymbol{U}_{jz} = IoU(\boldsymbol{b}_j^{gt}, \boldsymbol{p}_z),$$

where $(\boldsymbol{q}_j^{gt}, \boldsymbol{b}_j^{gt}) \in \Gamma^{gt}$, the best bounding box proposal $\boldsymbol{p}_{j*}$ is defined as:

$$j* = \operatorname*{argmax}_{z \in [1, \dots, e]} \boldsymbol{U}_{jz},$$

and the entries $(j \in [1, \dots, m], z \in [1, \dots, e])$ of matrix $\boldsymbol{C}$ containing the cosine similarity scores among the predicted class probabilities of the bounding box proposals as:

$$\boldsymbol{C}_{jz} = Sim\left(Pr_{Cls}(\boldsymbol{p}_{j*}), Pr_{Cls}(\boldsymbol{p}_z)\right),$$

where Sim is the cosine similarity function. Given these definitions, the entries of the target probability $\boldsymbol{P}^{target}$ is defined as:

$$\boldsymbol{P}_{jz}^{target} = \frac{\boldsymbol{U}_{jz}^*}{\sum_{i=1}^{e} \boldsymbol{U}_{ji}^*},$$

with:

$$\boldsymbol{U}_{jz}^* = \begin{cases} \boldsymbol{U}_{jz}\boldsymbol{C}_{jz}, & if\ \boldsymbol{U}_{jz} \geq \eta \\ 0, & \text{otherwise} \end{cases},$$

and η is a predefined threshold, i.e., a hyper-parameter.

On the basis of the above definitions, the grounding loss is defined as:

$$\begin{aligned} \mathcal{L}_g(\boldsymbol{P}, \mathcal{P}_{\boldsymbol{I}}, \Gamma^{gt}) &= \frac{1}{m}\sum_{j=1}^{m} KL_{div}(\boldsymbol{P}_j || \boldsymbol{P}_j^{target}), \\ &= \frac{1}{m}\sum_{j=1}^{m}\sum_{z=1}^{e} \boldsymbol{P}_{jz} \log\left(\frac{\boldsymbol{P}_{jz}}{\boldsymbol{P}_{jz}^{target}}\right), \end{aligned}$$

where KL_{div} is the KL divergence function, $\boldsymbol{P}_j$ ($\boldsymbol{P}_j^{target}$) is the j-th row of $\boldsymbol{P}$ ($\boldsymbol{P}^{target}$), and $\boldsymbol{P}_{jz}$ is the model predicted probability that the noun phrase $\boldsymbol{q}_j \in Q$ refers to the image content localized by $\boldsymbol{p}_z \in \mathcal{P}_{\boldsymbol{I}}$.

Indeed, the grounding loss captures both the bounding box spatial information and the semantic information determined by the bounding box classes. Whenever a bounding box is located near the ground truth bounding box and its class probability distribution is similar to the one of the best proposal $\boldsymbol{p}_{j*}$, then the loss favors the prediction of the bounding box; otherwise, the loss penalizes the bounding boxes according to their different probability distribution and spatial location. Previous works exploiting the KL divergence aim to maximize the probability of a bounding box proposal just considering their spatial location.

Now, the novel refinement loss will be defined. In order to do that, given a query $\boldsymbol{q}_j$, the following subset $\mathcal{S}_j \subseteq \mathcal{P}_{\boldsymbol{I}}$ of proposals need to be defined as:

$$\mathcal{S}_j = \{\boldsymbol{p}_z \mid \boldsymbol{p}_z \in \mathcal{P}_{\boldsymbol{I}} \wedge \boldsymbol{U}^*_{jz} \geq 0\},$$

which allows to define the loss $\mathcal{L}_c$ as:

$$\mathcal{L}_c(\mathcal{P}_{\boldsymbol{I}}, \Gamma^{gt}) = \frac{1}{m} \sum_{j=1}^{m} \sum_{\boldsymbol{p}_z \in \mathcal{S}_j} \hat{\boldsymbol{U}}_{jz} \mathcal{L}_{CIoU}(\boldsymbol{p}_z, \boldsymbol{b}^{gt}_j),$$

where $(\boldsymbol{q}^{gt}_j, \boldsymbol{b}^{gt}_j) \in \Gamma^{gt}$, and

$$\hat{\boldsymbol{U}}_{jz} = \frac{\boldsymbol{U}^*_{jz}}{max_{z \in [1,e]} \boldsymbol{U}^*_{jz} + \epsilon},$$

in which ϵ is a small value added to avoid division by 0, and $max_{z\in[1,e]}$ is the maximum function applied along the indexes $z \in [1, e]$. Intuitively, for each bounding box proposal that overlaps with the ground truth (according to the parameter η), this loss refines the coordinates proportionally to the "semantic" of the bounding box. Note that adopting the normalized scores $\hat{\boldsymbol{U}}_{jz}$, the model does not penalize the loss on the best bounding box proposal $j*$.

Bear in mind that this is the first work that proposes the exploitation of the probabilities distributions over the object detector classes to address the supervised visual-textual grounding task. However, in weakly-supervised visual-textual grounding, some approaches such as [89] leverage the information of the bounding box class with the *highest* probability.

5.4 Experimental Assessment

The model presented in this chapter is evaluated on two widely adopted datasets (i.e., ReferIt and Flickr30k Entities), considering several competing approaches in the literature, including State-of-the-Art models. In addition to that, in order to prove the usefulness of the proposed losses independently of the presented model architecture, the losses presented in this chapter are also adopted in the DDPN model. The choice of this model was due to: (i) publicly available code[2]; (ii) published results on both Flickr30k Entities and ReferIt datasets, with State-of-the-Art results on ReferIt; and (iii) exploitation of the same object detector used in this work.

5.4.1 Datasets and Evaluation Metric

Flickr30k Entities and ReferIt constitute the two most common datasets used in the literature, although other datasets have been used (e.g., [106, 107, 108, 109]). The Flickr30k Entities [77] dataset contains 32K images, 275K bounding boxes, 159K sentences, and 360K noun phrases. The ReferIt [110] dataset contains 20K images, 99K bounding boxes, and 130K noun phrases. See Appendix A.1 for more details about the Flickr30k Entities dataset and Appendix A.2 for more details about the ReferIt dataset.

For Flickr30k Entities, it is used the standard split for training, validation, and test set as defined in [77], consisting of 30K, 1K, and 1K images, respectively. For ReferIt, it is used 9K images of training, 1K images of validation, and 10K images of test.

Following all work in the literature, if a noun phrase corresponds to multiple ground truth bounding boxes, the boxes are merged and their union region is used as its ground truth. On the contrary, a noun phrase with no associated bounding box was removed from the dataset.

Aligned with the works in the literature, the standard *Accuracy* metric is adopted. Given a noun phrase, it considers a bounding box prediction to be correct if and only if the intersection over union value between the predicted bounding box and the ground truth bounding box is at least 0.5. See Appendix A.3 for more details about the Intersection over Union metric.

[2] The official code has been adapted: https://github.com/XiangChenchao/DDPN.

5.4.2 Model Selection and Implementation Details

To evaluate the proposed model on the test set of Flickr30k Entities and ReferIt datasets, it is selected the epoch in which the model achieved the best *Accuracy* metric on the validation set. A grid search for the best hyper-parameters, mainly for the Flickr30k Entities dataset, has been performed with the exception of the losses hyper-parameters visible in Section 5.4.3.2. For the ReferIt dataset, the other hyper-parameters values selected on the Flickr30k Entities dataset are used. The Adam optimizer is used, with the exponential learning rate scheduler set to 0.9, and the following values for the learning rate: $\{0.05, 0.03, 0.01, 0.005, 0.001\}$, c : $\{2048, 2053, 2060\}$, and η : $\{0.1, 0.3, 0.4, 0.45, 0.5, 0.55\}$. Other hyper-parameters are fixed to single values. For the textual features: $w = 300$, $t = 500$, and the LSTM network uses only one hidden layer of dimension t. For the image features, a fixed number $e = 100$ of proposals, with size $v = 2048$, are extracted from the ResNet-101's layer *pool5_flat* for each image, and $s = 5$. The best model *Accuracy* is achieved in both datasets at epoch 9 of training with a learning rate set to 0.001 and $c = 2053$. For Flickr30k Entities, $\eta = 0.3$ and $\lambda = 1$, while for ReferIt $\eta = 0.5$ and $\lambda = 1.4$. The code is publicly available on GitHub [3]. See Appendix B.2 for more details about the implementation of the model proposal.

5.4.3 Results

Table 5.1 reports the results obtained on the Flickr30k Entities dataset by the approach presented in this chapter and many other approaches presented in the literature, including the most recent State-of-the-Art models reported at the bottom part of the table. Concerning the model CMGN developed in [116], for the sake of a fair comparison, it is reported the performance obtained using the same setting of this chapter. In fact, the complete version of the CMGN model achieves an *Accuracy* of 76.74%, but exploiting query dependency information that the model presented in this chapter could exploit as well. The integration of this information into the proposed model is left for future work. It can be noted that the proposed approach significantly improves over competing approaches. Moreover, the DDPN model where the losses proposed in this chapter are used (last row of the table) shows a significant improvement in performance (1.03%) with respect to the original version.

Table 5.2 reports the results obtained on the ReferIt dataset by the proposed approach and the subset of the competing approaches reported in Table 5.1 that can be applied to

[3] `https://github.com/drigoni/Loss_VT_Grounding`

Model	Accuracy (%)
SCRC [82]	27.80
SMPL [111]	42.08
NonlinearSP [92]	43.89
GroundeR [6]	47.81
MCB [78]	48.69
RtP [77]	50.89
Similarity Network [112]	51.05
IGOP [113]	53.97
SPC+PPC [76]	55.49
SS+QRN [74]	55.99
SeqGROUND [114]	61.60
CITE [115]	61.89
QRC net [74]	65.14
YOLO [24]	68.69
DDPN [23]	73.30
CMGN [116]*	73.46
SL-CCRF [117]	74.69
The proposed model	**75.55**
DDPN [23] using the new losses	74.33

Table 5.1: Results obtained on Flickr30k test set. *Accuracy* indicates in percentage the standard accuracy metric. All values are copied from the original articles. "*" indicates that the reported model accuracy is referring to the version of the model in their ablation study, since the complete model uses query dependency information that is not exploited in this work.

this dataset, plus additional approaches that have been assessed on this dataset[4]. The model proposed in this chapter improves the *Accuracy* value by 3.02% when compared to the State-of-the-Art model (i.e., DDPN) for this dataset, representing a more significant gain than the one obtained on Flickr30k Entities. On the other hand, adopting the proposed losses in DDPN leads to the best performance, with an improvement over the original version of 3.66%. In the ReferIt dataset, each sentence corresponds to a single query independently from the others. In contrast, in Flickr30k Entities, a sentence could contain more queries that are semantically related among them. For this reason, models that apply complex multi-modal feature fusion components that aim to capture information among the queries extracted by the sentence in input sometimes do not consider the ReferIt dataset. Thus, the set of the models used as a comparison in the ReferIt dataset is not the same as in Flickr30k

[4] Some of them do not define an acronym, so the reference to the paper is used.

Model	Accuracy (%)
SCRC [82]	17.93
GroundeR [6]	26.93
MCB [78]	28.91
CITE [115]	34.13
IGOP [113]	34.70
[118]	36.18
QRC net [74]	44.10
[119]	44.20
YOLO [24]	59.30
DDPN [23]	63.00
The proposed model	66.02
DDPN [23] using the new losses	**66.66**

Table 5.2: Results obtained on ReferIt test set. *Accuracy* indicates in percentage the standard accuracy metric. All values are reported from the original articles.

Entities, and these reasons could explain the higher gain in *Accuracy* obtained in ReferIt than Flickr30k Entities.

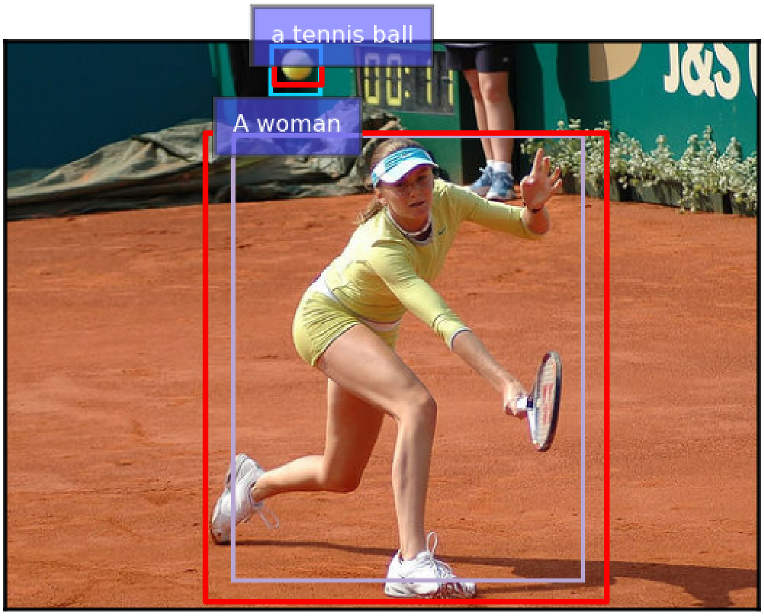
The *Point Game Accuracy* is also reported, which is recently used for a few models addressing the weakly-supervised task. It considers a prediction to be correct if and only if the center of the predicted bounding box is contained in the ground truth bounding box. In particular, the proposed model obtains 87.96% and 78.0% on Flickr30k Entities and ReferIt, respectively. These values are far better than the ones reported in the literature, and they suggest that a significant subset of predictions that are considered to be wrong according to the *Accuracy* metric, still refer to bounding boxes that have a significant overlap with the ground truth.

More information about the computational complexity of the approaches considered in this work is reported in Appendix B.3.

According to further experiments performed in Section 7.1 regarding the classes adopted by the object detector considered in this work, i.e., the Bottom-Up Faster R-CNN [28], these contains visually equivalent categories such as “lady” and “woman”. These equivalent classes share similar embedding features and similar probability distribution, as whenever the model needs to predict a category for an object appearing in the image, the model needs to split its predicted probabilities among all equivalent categories. The loss proposed in this chapter, which is based on the similarity among class probabilities, may grasp these equivalent classes and penalize them less during training. Further experiments in this direction are left as future works.

5.4.3.1 Qualitative Results

Figures 5.2,5.3, and 5.4 show qualitative examples predicted by the model proposed in this chapter on the test set of both Flickr30k Entities and ReferIt datasets. When the query refers to a small object in the image, most of the time, the model predicts a very close bounding box, but not enough to have the IoU score over the 0.5 value. This is the case for the query "a tennis ball" in the figure 5.2. See Appendix B.4 for more examples of qualitative results.

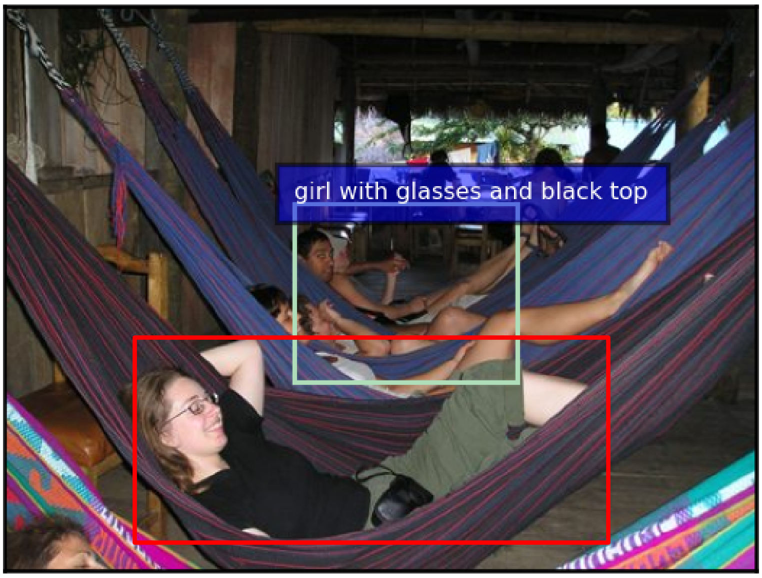


Sentence: "A woman tries to volley a tennis ball".

Figure 5.2: This picture reports a qualitative example of the proposed approach on the Flickr30k test image id: 23016347. The ground truth bounding boxes associated with each query are reported in **red**. The prediction for the query "a tennis ball" is evaluated as wrong, even if the bounding box is very close to the ground truth.

5.4.3.2 Ablation Study

The loss presented in this chapter is composed of two main components and two hyper-parameters. Here, it is reported the contribution of each part of the loss using different

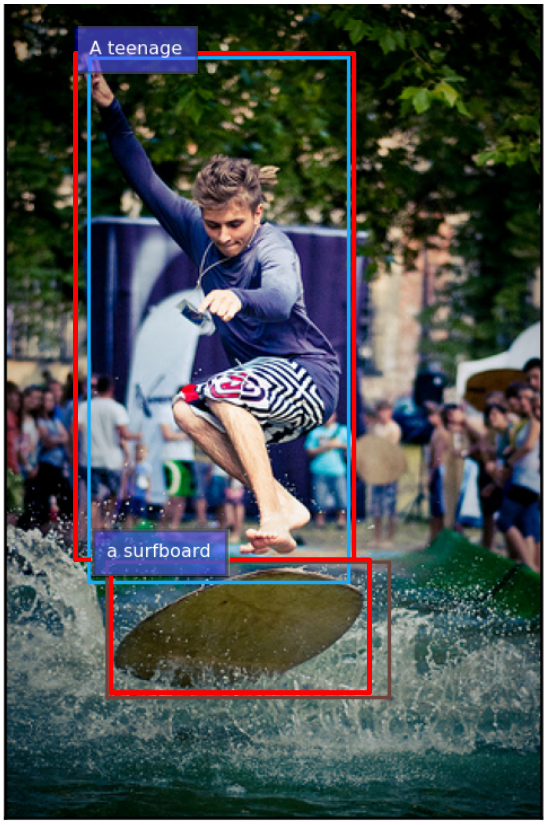


Sentence: "girl with glasses and back top".

Figure 5.3: This picture reports a qualitative example of the proposed approach on the ReferIt test image id: 14651. The ground truth bounding box is reported in **red**. The complete sentence in input is reported at the bottom of the figure. The predicted bounding box presents an intersection over union value with the ground truth of 0.08.

hyper-parameters values. A set of experiments are performed, where the grounding component is alternatively the cross-entropy, the KL divergence, or the proposed semantic KL divergence, and the regression component is alternatively the Smooth L1 or the proposed semantic CIoU. Moreover, different values for the hyper-parameters are considered. The obtained results (Table 5.3) show that the major contribution to the improvement is given by the *Complete IoU loss with semantic information*, which improves the model *Accuracy* by $\sim 2.6\%$ and $\sim 3.9\%$ on Flickr30k Entities and ReferIt datasets, respectively. Significant improvements are also obtained by using the semantic KL divergence in place of cross-entropy or the CIoU-Sem instead of the standard CIoU. Moreover, results show that the proposed approach is not much sensitive with respect to the hyper-parameters values[5], and, more importantly, the *Accuracy* on the validation set indeed represents well the *Accuracy* on the test set on both datasets.

[5] For new datasets, $\lambda = 1$ and $\eta = 0.5$ are good starting points, although model selection may result in better values.



Sentence: "A teenage is on a surfboard".

Figure 5.4: This picture reports a qualitative example of the proposed approach on the Flickr30k test image id: 6059154572. The ground truth bounding boxes associated with each query are reported in **red**. The complete sentence in input is reported at the bottom of the figure. All bounding boxes are predicted correctly.

5.5 Related Works

This proposed work is related mainly to two areas of research, namely, Visual-Textual Grounding and Visual-Textual-Knowledge Entity Linking (VTKEL). Moe details regarding both the Visual-Textual Grounding and the VTKEL problems are presented in Chapter 3. The work presented in this chapter is the first to adopt the CIoU loss in order to refine the final bounding boxes coordinates. Moreover, it is the first that extends the coordinates refinement to the subset of proposals that significantly overlap the ground truth, modulating the refinement by the agreement between the class probability of the best proposal and the class probability of the considered proposal.

5.6 Conclusion and Feature Work

This chapter introduced the first contribution of this Ph.D. thesis regarding the resolution of the traditional Visual-Textual Grounding task. More in detail, it introduced a novel loss jointly with a simple two-stage approach model. The novel loss combines a grounding loss and a bounding box coordinates refinement loss, both based on semantic information, i.e., a probability distribution over a set of pre-defined classes, returned by the object detector. The experimental assessment showed that the proposed approach was able to reach a higher accuracy than State-of-the-Art models, even without using a more complex multi-modal feature fusion component. Specifically, the proposed model results are compared to several models in the literature over two commonly used datasets, Flickr30k Entities, and ReferIt. With respect to the best State-of-the-Art approaches, on the Flickr30k Entities dataset, the proposed approach obtained an improvement of 0.86%, while on the ReferIt dataset, it improved the State-of-the-Art performance by 3.02%. Applying the proposed loss to the DDPN model significantly improves its performance on both datasets, demonstrating the proposed loss usefulness independently from the proposed model.

Since the model proposed in this chapter uses a simple multi-modal feature fusion component, there is space for trivial improvements, including a more sophisticated multi-modal feature fusion component, such as bilinear-pooling [78] and deeper architectures, as well as the exploitation of dependencies among the queries contained by the input sentence. Future work will also address more sophisticated object detectors and the idea of including different forms of information, such as a scene graph and prior knowledge.

Losses		Hyper-par.		Flickr30k (%)		ReferIt (%)	
Gr.	Reg.	λ	η	Val.	Test	Val.	Test
CE	SmoothL1	0.8	/	**71.25**	**71.82**	64.24	**61.81**
		1	/	71.08	71.61	64.19	61.29
		1.2	/	71.18	71.21	**64.65**	61.64
KL	SmoothL1	0.8	0.4	71.51	72.06	63.58	61.38
		0.8	0.5	72.16	**72.55**	64.57	**62.69**
		1	0.4	71.76	72.34	63.93	61.65
		1	0.5	**72.58**	72.18	**64.82**	62.49
KL-Sem	SmoothL1	0.8	0.4	72.22	72.72	64.38	61.78
		0.8	0.5	72.42	72.41	64.99	62.12
		1	0.4	**72.54**	**72.88**	65.04	62.47
		1	0.5	72.34	72.83	**65.45**	**62.72**
CE	CIoU-Sem	0.8	0.4	73.99	74.56	**67.66**	**65.47**
		0.8	0.5	73.60	74.24	67.41	65.07
		1	0.4	**74.07**	**74.82**	67.60	65.42
		1	0.5	73.90	74.24	67.24	65.15
KL-Sem	CIoU-Sem	0.6	0.5	75.17	75.38	68.23	66.31
		0.8	0.5	75.27	**75.67**	68.70	66.12
		1	0.5	75.41	75.53	68.72	66.52
		1.2	0.5	75.23	75.34	68.88	66.37
		1.4	0.5	75.13	75.36	**68.97**	66.02
		1	0.3	**75.60**	75.55	68.64	66.49
		1	0.4	75.40	75.64	68.56	**66.54**
		1	0.6	74.48	74.68	68.02	65.31

Table 5.3: Accuracy obtained on Flicker30k Entities and ReferIt datasets as the losses functions and hyper-parameters values change. *CE* indicates the cross-entropy loss, *SmoothL1* indicates the Smooth L1 loss, *KL-Sem* indicates the KL loss with the semantic information and *CIoU-Sem* indicate the Complete IoU loss with the semantic information. The baseline model does not use the η parameter.

6

Weakly-Supervised Visual-Textual Grounding with Semantic Prior Refinement

As presented in the introduction of this Ph.D. thesis, while developing the probabilistic framework, the traditional visual-textual grounding task which considers only two modalities (i.e., image and text) was also studied. The previous Chapter 5 presented a model with a new loss to solve the visual-textual grounding task in a supervised setting. Instead, this chapter[1] proposes a new approach for solving the visual-textual grounding in the weakly-supervised setting. Bear in mind that during model training there is no information available about the location of the object in the image, nor the ground truth alignment between queries and bounding boxes. Thus, the loss presented in Chapter 5 cannot be applied in this setting.

This chapter proposes a simple model dubbed Semantic Prior Refinement (SPR) model, whose predictions are obtained by combining the output of two main modules: (i) the first module, which does not require learning, aims to return, for each textual phrase, a rough alignment with the corresponding bounding box referred by the phrase; (ii) the second module, composed by two sub-components which do require learning, refines the rough predic-

[1] **D. Rigoni**, L. Parolari, L. Serafini, A. Sperduti, and L. Ballan, "Weakly-Supervised Visual-Textual Grounding with Semantic Prior Refinement", *Under Peer Review*.

tion in the final phrase-bounding box alignments. The model is trained to maximize the multimodal similarity between an image and a sentence while minimizing the multimodal similarity of the same sentence and a new unrelated image, carefully selected to help the most during training. The model performances on the Flickr30k Entities and the ReferIt datasets are investigated. The proposed approach presents State-of-the-Art results in both datasets. Moreover, thanks to the untrained component, it reaches competitive performances just using a small fraction of training examples.

6.1 Introduction

Visual-textual grounding, i.e., the task of finding region-phrase correspondences, requires a joint understanding of both visual and textual modalities. Despite the outstanding advancements in computer vision and natural language processing, it remains a hard task. According to the amount of dataset annotations used for training the model, this task can be tackled in a fully supervised or weakly-supervised manner. In the first setting, the model is trained using all the region-phrase pairs [6, 7, 8, 9, 10, 11], while in the second setting [87, 88, 9, 89, 85] the only available annotation refers to image-sentence pairs. In other words, it is known only which sentence describes each image in the dataset, but not the objects in the image referred by the textual phrases composing the sentence. The differences between these two approaches are visible in Figure 1.2. Under the fully supervised setting, the model requires knowing the bounding box referred by the textual phrase, an extremely hard and expensive annotation to collect. For this reason, this chapter focuses on solving the visual-textual grounding task under a weakly-supervised setting.

A simple model is proposed, dubbed Semantic Prior Refinement (SPR) model, whose predictions are obtained by combining two modules: (i) the first, which does not require training, for each textual phrase returns a rough alignment with the bounding box referred by the phrase, while (ii) the second, composed by two trained sub-components, refines the rough predictions in the final phrase-bounding box alignments. Given a textual phrase and an image as input, the model recognizes the most relevant objects in the image using a pre-trained object detector and predicts as output the bounding box referred by the phrase. Specifically, the rough alignment is based on the similarity score (i.e., concept similarity) between the head of the textual phrase and the predicted label of the bounding boxes. Here, the key idea is that the head of the phrase should be very similar (semantically speaking) to the content of the bounding box and, thus, to its class.

The model is trained to maximize the multimodal similarity between an image and a sentence describing that image while minimizing the multimodal similarity of the same sentence and a new unrelated image, adequately selected. The model performances on the Flickr30k Entities and the ReferIt datasets are investigated, showing that it presents consistent and competitive results in both datasets. Moreover, the model performance is evaluated in low-data environments, showing that it can still achieve surprising results even when trained with just a tiny fraction of training examples

The main contributions can be summarized as follows: (i) it is proposed a new model which is based on the novel idea of first predicting a rough alignment between the phrase and a bounding box, and then refining the prediction; (ii) extensive experiments are conducted on the popular Flickr30k Entities and ReferIt datasets, showing state-of-the-art results (in the weakly-supervised setting); (iii) the proposed approach, even when trained on a small fraction of the available examples (e.g., 10%), achieves consistently competitive results.

6.2 Problem Definition

The work presented in this chapter tackles the visual-textual grounding task whose formal definition is introduced in Section 4.1. Following the setting adopted in Chapter 5, it is assumed that each noun phrase is associated with one and only one bounding box, while a bounding box, can identify more objects.

Bear in mind that in the weakly-supervised approach, the training set is defined as $\mathcal{D} = \{(\boldsymbol{I}_i, \mathrm{S}_i)\}_{i=1}^{n}$, where n is the number of examples. In other words, during model training, only the information about sentence S_i describing the image $\boldsymbol{I}_i$ is available, while there is no information about which noun phrase $\boldsymbol{q} \in \mathcal{Q}_{\mathrm{S}}$ refers to each bounding box $\boldsymbol{b} \in \mathcal{B}_{\boldsymbol{I}}$ (i.e., Γ_i^{gt}).

In this work, given an image $\boldsymbol{I}$, a pre-trained object detector is deployed to extract the set of bounding box proposals $\mathcal{P}_{\boldsymbol{I}} = \{(\boldsymbol{c}_k, \boldsymbol{h}_k, \boldsymbol{l}_k)\}_{k=1}^{p} \subset \mathcal{B}_{\boldsymbol{I}}$ that should contain all the objects depicted in the image $\boldsymbol{I}$, where $\boldsymbol{c}_k \in \mathbb{R}^4$ represents the coordinates of the bounding box located in the image, $\boldsymbol{h}_k \in \mathbb{R}^v$ is the v-dimensional vector representing the bounding box features, and $\boldsymbol{l}_k \in \Theta$ denotes the class with the highest probability (over the object detector pre-defined set of categories Θ) that best represents the content of the bounding box. The features are the internal object detector activation values of the hidden layer just before the classification layers and the regression layer for the prediction of the bounding boxes coordinates. Classes information is typically returned by most of the object detectors, and it will be used in Sec. 6.3.1 to define the concept similarity.

Hence, given an image $\boldsymbol{I}$ with a set of bounding box proposals $\mathcal{P}_{\boldsymbol{I}}$ defined on $\boldsymbol{I}$, and a sentence $\mathrm{S} \in \mathcal{S}$ with a set of noun phrases $Q_{\mathrm{S}} \subseteq \mathcal{Q}_{\mathrm{S}}$ defined on S, then $\gamma(\boldsymbol{I}, S)$ returns a subset $\Gamma \subseteq Q_{\mathrm{S}} \times \mathcal{P}_{\boldsymbol{I}}$ where each couple $(\boldsymbol{q}, \boldsymbol{p}) \in \Gamma$ associates the noun phrase $\boldsymbol{q}$ to the bounding box proposal $\boldsymbol{p}$.

6.3 The Model Proposal

This section presents the model proposal architecture and the novel contribution based on the assumption of first predicting rough alignments between queries and bounding boxes, and then refining those predictions using a trained module.

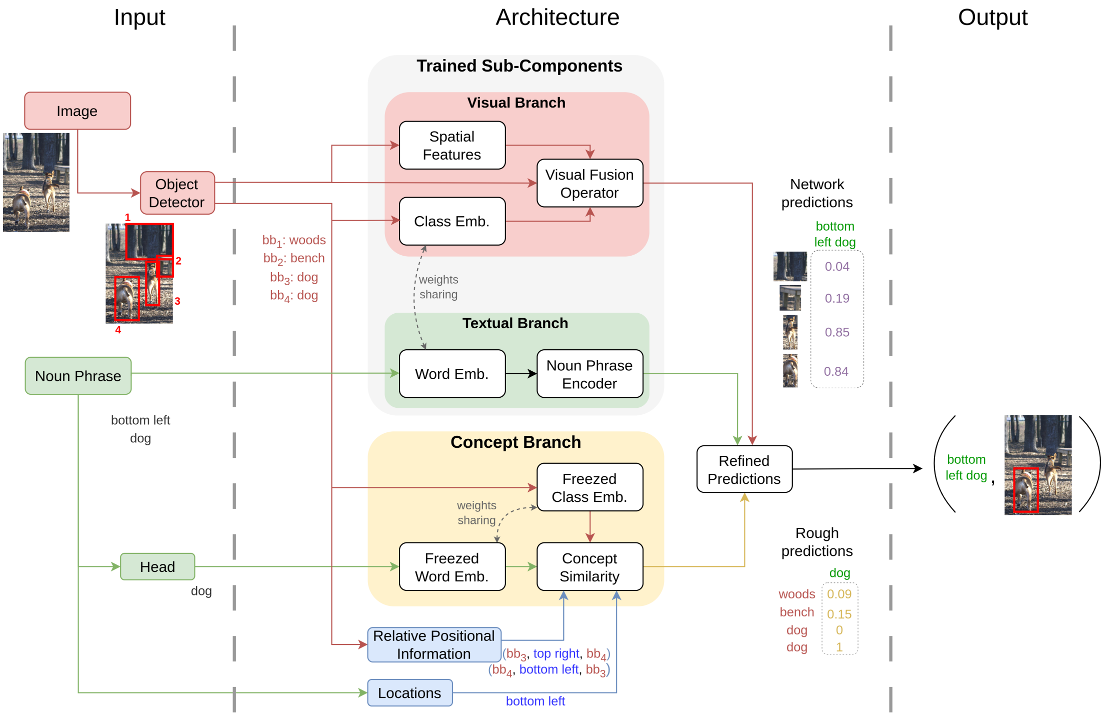


Figure 6.1: The model architecture overview. The model computes a first rough set of alignments by leveraging prior knowledge from the object detector and word embedding (i.e., *Concept Branch*). A simple positional heuristic is injected as an extra source of prior knowledge to reduce ambiguity for candidate alignments. Then, the visual and textual branches (i.e., *Trained Sub-Components*) match learned multimodal features to predict a second, refined set of alignments. The two sets are then combined together by the *Refined Predictions* module to compute final scores for grounding.

Figure 6.1 depicts the Semantic Prior Refinement (SPR) model architecture, which is composed mainly of two modules. One is the *Concept Branch* (see Section 6.3.1), responsible for predicting a first rough set of region-phrase correspondences. Those alignments are obtained through a process named “concept similarity” that captures the semantic information conveyed by prior knowledge in object detector and word embedding. In particular, it compares the word embeddings of the phrase’s head and the bounding box class to get unimodal scores. No training is required. The information is matched by relying on two important assumptions: (i) the proposal’s label semantically describes the bounding box content, (ii) and the word embedding space represent the semantic similarity of the words. In addition, the *Concept Branch* incorporates a simple positional heuristic that helps to reduce ambiguity for candidate alignments.

The other module (see Section 6.3.2) is made by two sub-components, namely *Visual Branch* and *Textual Branch*, and it is trained to learn a multimodal embedding space for region-phrase correspondences given image-sentence pairs. The multimodal representations are constructed to maximize the similarity of region-phrase pairs when both come from the same example while minimizing the similarity between the regions from the positive example and phrases from another example. The second refined set of alignments is obtained by measuring the similarity between learned multimodal visual and textual features for the bounding box proposal and noun phrase. The resulting scores are then combined by the prediction refinement module (see Section 6.3.3) to produce final scores. The candidate alignment is chosen to be the proposal with maximum similarity with the noun phrase.

6.3.1 Concept Branch

The *Concept Branch* (CB) is designed to face the most important problem in the weakly-supervised visual grounding: the unavailability of region-phrase ground truths. The idea is to use external sources of knowledge to fill this gap. The CB leverages a pre-trained object detector to abstract the content of an image’s region through the bounding box classification label, that is the concept expressing the content of the region. The bounding box classification label is a common feature in most object detectors allowing them to express the content of the bounding box as a concept in the language domain. To understand the concept expressed by a textual phrase, an off-the-shelf NLP parser is deployed to extract the head of the phrase [84]. In fact, the head of a textual phrase determines its syntactic category. Then, by means of a pre-trained word embedding that convey prior knowledge on words, the CB computes the similarity between the two concepts to obtain a rough score named “concept similarity”. There is no training involved in this process; thus, the process is entirely independent of training data and can be treated as prior knowledge.

Although general enough to cover a vast set of cases, this method has some limitations. First, the proposal's classification may be noisy and incorrect, driving the CB to inaccurate alignments. Second, the word embedding similarity may be biased and imprecisely capture the semantic similarity between words. Third, the CB produces equal scores when proposals have the same label. See Appendix C.3 for more details about these limitations. In order to deal with this issue, another source of prior knowledge based on spatial relations is adopted. For proposals with the same label, relative positional information are extracted (e.g., top, left, etc.). Then the relations are matched with a location extracted from the phrase by a simple text search (e.g., "left" in "the woman on the left").

Formally, given a set of p bounding box proposals $\mathcal{P}_{\boldsymbol{I}}$, let $E^{\mathcal{P}_{\boldsymbol{I}}} = \{\boldsymbol{e}_k^{\mathcal{P}_{\boldsymbol{I}}}\}_{k=1}^{p}$ be the corresponding set of g-dimensional vectorial embeddings, where each $\boldsymbol{e}_k^{\mathcal{P}_{\boldsymbol{I}}}$ is the embedding of the bounding box class $\boldsymbol{l}_k$, for $1 \leq k \leq p$. Given a noun phrase $\boldsymbol{q}_j$ composed by a sequence of $L(\boldsymbol{q}_j)$ words $W^{\boldsymbol{q}_j} = [w_1^{\boldsymbol{q}_j} \dots w_{L(\boldsymbol{q}_j)}^{\boldsymbol{q}_j}]$, let $E^{\boldsymbol{q}_j} = \{\boldsymbol{e}_i^{\boldsymbol{q}_j}\}_{i=1}^{L(\boldsymbol{q}_j)}$ be the set of words embedding of size g associated with each word in the noun phrase $\boldsymbol{q}_j$. The model also keeps track of positional information. Inspired by [120], let $\boldsymbol{s}_j^t \in \mathbb{R}^6$ and $\boldsymbol{s}_k^v \in \mathbb{R}^6$ be two multi-hot vectors that encode locations in $\boldsymbol{q}_j$ and relations in the k-th proposal, respectively (more details in Section 6.4.2).

Then, the concept similarity score for each proposal is calculated as follows:

$$\boldsymbol{S}_{jk} = f_{mask}\left(\boldsymbol{\xi}_j, \boldsymbol{e}_k^{\mathcal{P}_{\boldsymbol{I}}}, \boldsymbol{s}_j^t, \boldsymbol{s}_k^v\right) = \begin{cases} f_{\text{sim}}\left(\boldsymbol{\xi}_j, \boldsymbol{e}_k^{\mathcal{P}_{\boldsymbol{I}}}\right) & \text{if } \left(\boldsymbol{s}_k^v\right)^\top \boldsymbol{s}_j^t \geq 0; \\ -1 & \text{otherwise.} \end{cases}$$

where f_{sim} is a similarity measure (such as the cosine similarity), and $\boldsymbol{\xi}_j$ is the word embedding of the head of the noun phrase $\boldsymbol{q}_j$. f_{mask} is a masking function that, whenever a spatial reference is in the noun phrase, selects only the bounding boxes that are in the spatial position indicated by the noun phrase in the image. In other words, if the word "bottom" appears in the noun phrase, then f_{mask} penalizes the bounding boxes in the middle and top regions of the image. The new embedding $\boldsymbol{\xi}_j$ is the average of the word embedding representations of the phrase's head which are extracted using an NLP parser. Here, the key idea is to consider only the most meaningful words which compose the textual phrase and to avoid the inclusion of words that do not carry more meaning but are used to structure the sentence, such as verbs and prepositions. Note that this part of the model's architecture does not require training.

6.3.2 Visual and Textual Branches

Given the set of bounding box proposals $\mathcal{P}_{\boldsymbol{I}}$ detected in the image $\boldsymbol{I}$ by the object detector, for each of them, the proposed model calculates the spatial features $H^s = \{\boldsymbol{h}_k^s\}_{k=1}^p$ where $\boldsymbol{h}_k^s \in \mathbb{R}^5$, as indicated in [10].

In contrast to the Concept Branch, the Visual and Textual branches adopt trainable word embeddings $\overline{E}^{\mathcal{P}_{\boldsymbol{I}}} = \{\overline{\boldsymbol{e}}_k^{\mathcal{P}_{\boldsymbol{I}}}\}_{k=1}^p$ and $\overline{E}^{\boldsymbol{q}_j} = \{\overline{\boldsymbol{e}}_i^{\boldsymbol{q}_j}\}_{i=1}^{L(\boldsymbol{q}_j)}$ associated to the bounding box classes and to the words of the noun phrases, respectively. Initially, both visual and spatial features are concatenated and then projected on a smaller dimensional space, thus leading to a set of new vectorial representations $H^{||} = \{\boldsymbol{h}_k^{||}\}_{k=1}^p$, with $\boldsymbol{h}_k^{||} = \boldsymbol{W}^{||}\big(\boldsymbol{h}_k^s||\boldsymbol{h}_k\big) + \boldsymbol{b}^{||}$, where $||$ indicates the concatenation operator, $\boldsymbol{h}_k^{||} \in \mathbb{R}^g$, $\boldsymbol{W}^{||} \in \mathbb{R}^{g\times(5+v)}$ is a matrix of weights, and $\boldsymbol{b}^{||} \in \mathbb{R}^g$ is a bias vector. The new representation is then summed to the word embedding of the bounding box label to obtain the final visual features $\boldsymbol{h}_k^v = \boldsymbol{h}_k^{||} + \overline{\boldsymbol{e}}_k^{\mathcal{P}_{\boldsymbol{I}}}$, where $\boldsymbol{h}_k^v \in \mathbb{R}^g$.

Given the set $\overline{E}^{\boldsymbol{q}_j}$ of trainable word embeddings associated with the noun phrase $\boldsymbol{q}_j$, the textual branch applies a function f_{enc} to generate only one embedding $\boldsymbol{h}_j^t \in \mathbb{R}^\tau$ for each phrase $\boldsymbol{q}_j$. This textual features extraction is defined as $\boldsymbol{h}_j^t = f_{enc}(\overline{E}^{\boldsymbol{q}_j})$.

Note that the embeddings $\overline{E}^{\mathcal{P}_{\boldsymbol{I}}}$ and $\overline{E}^{\boldsymbol{q}_j}$ are generated with trainable modules that share the weights among each other (weights sharing). So, during training, the word embeddings learn multimodal embeddings for the visual and textual information.

6.3.3 Refined Predictions

The prediction module is in charge of refining the rough predictions $\boldsymbol{S}_{jk}$, i.e., the *Concept Branch* predicted scores, using the visual $\boldsymbol{h}_k^v$ and textual $\boldsymbol{h}_j^t$ features. Initially, starting from $\boldsymbol{h}_k^v$ and $\boldsymbol{h}_j^t$, the model predicts the probability $\boldsymbol{P}_{jk}$ that a bounding box proposal of index k is referred by the noun phrase $\boldsymbol{q}_j$ as $\boldsymbol{P}_{jk} = f_{\text{sim}}(\boldsymbol{h}_k^v, \boldsymbol{h}_j^t)$, where f_{sim} is a similarity measure between vectors. Please note that in this work, the cosine similarity function is adopted; therefore, $\boldsymbol{h}_k^v$ and $\boldsymbol{h}_j^t$ have the same vector dimension, i.e., $g = \tau$.

Finally, the rough predictions are refined via the scores $\boldsymbol{P}_{jk}$ using an hyperparameter $\omega \in \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ as:

$$\hat{\boldsymbol{P}}_{jk} = \omega * \boldsymbol{P}_{jk} + (1-\omega) * \boldsymbol{S}_{jk}.$$

The major benefit of this approach is that model predictions are not constrained to values defined by concept similarity: they co-work for the final predictions.

6.3.4 Loss Function

Inspired by [88], this work adopts a contrastive loss. The contrastive objective $\mathcal{L}$ aims to learn the visual and textual features by maximizing the similarity score between paired image-sentence examples and minimizing the score between the negative examples.

Formally, given two training examples $(\boldsymbol{I}, \mathrm{S}), (\boldsymbol{I}', \mathrm{S}') \in \mathcal{D}$ such that $\mathrm{S} \neq \mathrm{S}'$ and $\boldsymbol{I} \neq \boldsymbol{I}'$, the loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = -\underbrace{f_{pair}(\boldsymbol{I}, \mathrm{S})}_{\text{Positive example}} + \underbrace{f_{pair}(\boldsymbol{I}', \mathrm{S})}_{\text{Negative example}},$$

where f_{pair} is the similarity function defined over the multimodal pair image-sentence, defined as:

$$f_{pair}(\boldsymbol{I}, \mathrm{S}) = \frac{1}{m} \sum_{j=1}^{m} \max_{k} \frac{\hat{\boldsymbol{P}}_{jk}}{\sum_{i}^{p} \hat{\boldsymbol{P}}_{ji}}$$

where m is the number of queries in S and $\hat{\boldsymbol{P}}_{jk}$ is the predicted similarity between the noun phrase $\boldsymbol{q}_j$ and proposal $\boldsymbol{p}_k$. Basically, the goal of f_{pair} is to aggregate the similarity scores of all the region-phrase pairs, determining the degree to which the phrases correspond with the content of the image.

In contrast to what is done in [88] where for each positive example, several negative examples built from the batch are considered, this proposed approach adopts just a specific negative example $(\boldsymbol{I}', \mathrm{S})$. The negative example is built from the example $(\boldsymbol{I}', \mathrm{S}')$, selected from the batch precisely to be the one where the sentence S' is the most similar to the sentence S. This allows the model to focus on fine-grained region-phrase details that differ between the two examples. Specifically, given a training example $(\boldsymbol{I}, \mathrm{S}) \in \mathcal{B}$, the negative example is selected as:

$$(\boldsymbol{I}', \mathrm{S}') = \underset{(\boldsymbol{I}'', S'') \in \mathcal{B}'}{\operatorname{argmax}} f_{sim}(\zeta(\mathrm{S}''), \zeta(\mathrm{S}))$$

$$\zeta(\mathrm{S}) = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{L(\boldsymbol{q}_j)} \sum_{i=1}^{L(\boldsymbol{q}_j)} \boldsymbol{e}_i^{\boldsymbol{q}_j}$$

where $\mathcal{B}' = \mathcal{B} \backslash \{(\boldsymbol{I}, \mathrm{S})\}$. In other words, the similarity among sentences is measured in the word embedding space.

6.4 Experimental Assessment

This section presents the model results and evaluation protocol. The proposed model results are evaluated considering several competing approaches in the literature, including State-of-the-Art models.

6.4.1 Datasets and Evaluation Metrics

In this work, the presented approach is evaluated on both the Flickr30k Entities [77] and ReferIt [107] datasets. The Flickr30k Entities dataset contains 32K images, 275K bounding boxes, 159K sentences, and 360K noun phrases. The ReferIt [107] dataset contains 20K images, 99K bounding boxes, and 130K noun phrases. For Flickr30k Entities, it is used the standard split for training, validation, and test set as defined in [77], consisting of 30K, 1K, and 1K images, respectively. For ReferIt it is used 9K images of training, 1K images of validation, and 10K images of test. See Appendix A.1 for more details about the Flickr30k Entities dataset and Appendix A.2 for more details about the ReferIt dataset. Following common practice, if a noun phrase corresponds to multiple ground truth bounding boxes, the boxes are merged, and their union is used as ground truth. A noun phrase with no associated bounding box was removed from the dataset.

Aligned with the works in the literature, the standard *Accuracy* metric is adopted. Given a noun phrase, it considers a bounding box prediction to be correct if and only if the *Intersection over Union (IoU)* value between the predicted bounding box and the ground truth bounding box is at least 0.5. See Appendix A.3 for more details about the *IoU* metric.

Moreover, the *Pointing Game Accuracy* is also calculated for comparison purposes [80, 83, 91, 121]. *Pointing Game Accuracy* considers an example to be positive whether the center of the predicted bounding box lies wherever inside the ground truth box.

6.4.2 Model Selection and Implementation

The model selected for evaluating the test set of the Flickr30k Entities and of the ReferIt datasets is chosen on the epoch that better performs in terms of *Accuracy* in the validation set. The best hyper-parameters on both Flickr30k Entities and ReferIt datasets are searched, independently on the considered fractions of training data $\{5\%, 10\%, 50\%, 100\%\}$ used for learning. It is selected 10^{-5} for the learning rate among $\{10^{-3}, 10^{-4}, 10^{-5}\}$, and GloVe [3] is adopted as word embeddings. In both cases, the word embedding dimension was set to $\tau = g = 300$. In this work, f_{enc} is implemented with a recurrent neural network. Both an RNN and an LSTM [105] neural network were considered, and the best performances were obtained for the latter with hidden layer(s) of 300 units and 1 layer between $\{1, 2\}$. The normalization of the bounding boxes' spatial features with the dimension of the image as done in [10] was also considered as a hyper-parameter. Normalization was selected for ReferIt and unnormalized spatial features for Flickr30k Entities. The vector $\boldsymbol{h}^t_j$ is the τ-dimensional LSTM output of the last word $w^{\boldsymbol{q}_j}_{L(\boldsymbol{q}_j)}$ in the noun phrase $\boldsymbol{q}_j$. The bounding box proposals $\mathcal{P}_{\boldsymbol{I}}$ are extracted with the Bottom-Up Attention [66] object detector with a confidence score of 0.1 for Flickr30k[2] Entities and 0.2 for ReferIt[3]. There are many misspell and confusion among class labels due to the automatic extraction of classes done by the object detector trained on the Visual Genome [32] dataset. For this reason, this work employs a spell checker[4] to fix the errors. It is adopted a language parser[5] to automatically extract the noun phrases' heads. The bounding box features have a dimension of $v = 2048$ and are extracted from the *pool5_flat* layer in ResNet-101. The cosine similarity is used as a similarity measure f_{sim} between vectors. The best model performances are obtained with a batch size of 16 between $\{16, 32\}$ and the prior $\omega = 0.4$ selected among $\{0.1, 0.25, 0.4, 0.5, 0.75, 0.9\}$. Vectors $\boldsymbol{s}^t_j$ and $\boldsymbol{s}^v_k$ are binary vectors of dimension 6, where each position in the vector has the following meaning: [left, right, center, top, bottom, middle]. Whenever a spatial location is present in the noun phrase $\boldsymbol{q}_j$, then the corresponding position in $\boldsymbol{s}^t_j$ is set to value 1 and the other to value 0. If no spatial location is present, then the vector is initialized with all values 1. The vector $\boldsymbol{s}^v_k$ is always initialized with the positions of the bounding boxes. In particular, the bounding boxes positions are set according to only the bounding boxes that share the same class.

All experiments were performed in a cluster equipped with A5000 24GB GPUs. Each experiment is trained on a single GPU and requires about 6 hours when the model is trained on ReferIt, and about 15 hours when trained on Flickr30k Entities. All the experiments required about 2400 GPU hours. The final model architecture is composed of about 241M parameters, of which about 240M parameters compose the word embedding vocabularies. Hence, the *Concept Branch* module is made of 120M frozen word embedding parameters.

[2] The same features of [88] are adopted.

[3] https://github.com/MILVLG/bottom-up-attention.pytorch

[4] https://pypi.org/project/pyspellchecker/

[5] SpaCy, version 3.4.1: https://spacy.io/

Model	Flickr30k Entities (%)		ReferIt (%)	
	Accuracy	P. Accuracy	Accuracy	P. Accuracy
Top-Down Visual Saliency [91]	-	50.1	-	-
KAC Net [7]	37.7	-	15.8	-
Semantic Self-Supervision [84]	-	49.1	-	40.0
Anchored Transformer [85]	33.1	-	13.6	-
Multi-level Multimodal [80]	-	57.9	-	48.4
Align2Ground [83]	11.5	71.0	-	-
Counterfactual Resilience [122]	48.66	-	-	-
Multimodal Alignment Framework (MAF) [88]	61.4	-	-	-
Contrastive Learning [9]	-	74.9	-	-
Grounding By Separation [87]	-	70.5	-	59.4
Relation-aware [121]	59.27	78.60	37.68	58.96
Contrastive Knowledge Distillation [123]	53.10	-	38.39	-
SPR + CLIP	56.89	77.06	40.99	57.48
SPR model	**62.20**	**80.68**	**48.04**	**62.40**

Table 6.1: Results on Flickr30k Entities and ReferIt test sets. *Accuracy* is the standard accuracy metric, while *P. Accuracy* is the pointing game accuracy metric, both reported in percentage. The best values are shown in bold. The SPR model presents State-of-the-Art values for both Flickr30k Entities and ReferIt datasets.

6.4.3 Experiments

The proposed model is compared to several approaches in the literature on the Flickr30k Entities and ReferIt datasets. The model performance, when trained only with a small number of training examples, is also assessed. Indeed, the untrained *Concept Branch* module should give stability to the model even when it is trained on a small training set as it should help to counter the overfitting trend that occurs with small datasets.

6.4.3.1 Full Training Set Scheme

This section reports the results obtained by the proposed model on whole datasets.

Table 6.1 compares the proposed model results to those of several approaches in the literature. The model presented in this chapter outperforms all other approaches on standard *Accuracy* and *Pointing Game Accuracy*. In particular, in the Flickr30k Entities, the model's improvements over the State-of-the-Art are $+0.8\%$ in Accuracy and $+2.08\%$ in P. Accuracy. While on ReferIt, the improvements are $+9.65\%$ and $+3\%$, respectively for both the metrics.

To assess the soundness of the presented approach, it is tested a variant of the model that replaces visual and textual branches, responsible for learning the multimodal embedding space,

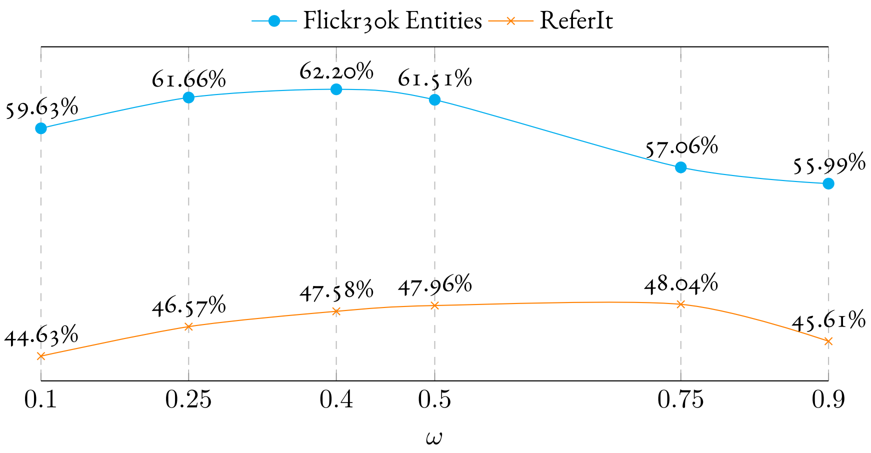


Figure 6.2: Accuracy results on Flickr30k Entities and ReferIt test sets varying the ω hyper-parameter. Results were obtained by training the model on 100% of the training set.

with CLIP's multimodal embeddings (referred to "SPR + CLIP") [124]. See Appendix C.1 for more details regarding the CLIP's Embeddings.

As the results show, in Table 6.1, the full SPR model still outperforms the variant with CLIP. This occurs because CLIP was trained to capture the multimodal coarse-grained information from image and sentence pairs, while in visual grounding, it is needed more fine-grained details regarding the alignments region-query.

The hyper-parameter ω regulates the weight of the *Concept Branch* on the final predictions: the higher the value, the less the *Concept Branch* affects final predictions. For this reason, Figure 6.2 presents the *Accuracy* results obtained with the SPR model using different values of ω: $\{0.1, 0.25, 0.4, 0.5, 0.75, 0.9\}$ when the model is trained on the entire training set. As shown by the chart, ω greatly affects the model performance in both datasets[6], allowing the model to reach its peak of performance when $\omega = 0.4$ in Flickr30k Entities and $\omega = 0.75$ in ReferIt.

More information about the computational complexity of the approaches considered in this work is reported in Appendix C.2.

6.4.3.2 Small Training Set Scheme

This section presents the results obtained with the SPR model on the datasets where only a fraction of training examples are used for training.

[6] For new datasets, $\omega = 0.4$ is a good starting point, although model selection may result in a better value since this parameter is very sensitive to the adopted dataset.

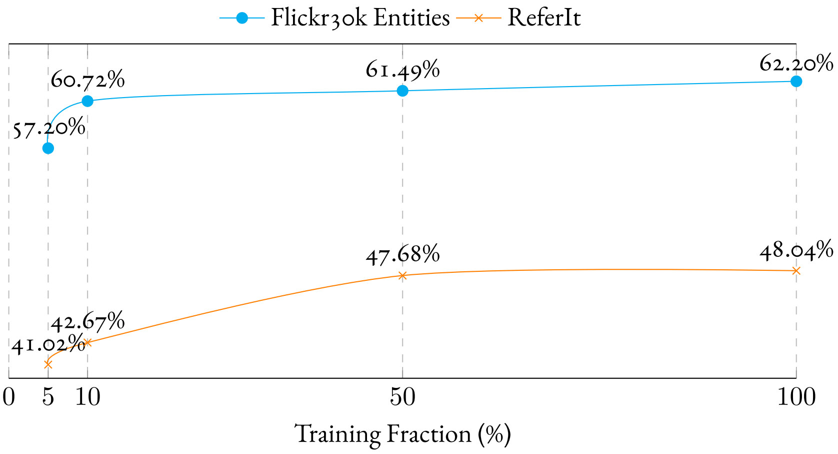


Figure 6.3: Accuracy results on Flickr30k Entities and ReferIt test set by the SPR model trained in low-data environments. The percentage refers to the fraction of the training set considered during training.

Concept Branch	Trained Modules	Rel. Posit. Information	Flickr30k Entities (%)	ReferIt (%)
✗	✓	✗	23.52	15.03
✓	✗	✗	54.96	40.07
✓	✗	✓	55.02	42.69
✓	✓	✗	62.10	45.44
✓	✓	✓	**62.20**	**48.04**

Table 6.2: Model Ablation. Accuracy of the model's components. The *Concept Branch* contributes more to the final model performances.

Figure 6.3 reports the SPR model *Accuracy* results. On Flickr30k Entities, the model is able to obtain State-of-the-Art results even when trained with only 50% of the training data, while on ReferIt, even when the model is trained with 5% of the training examples, it achieves State-of-the-Art performances.

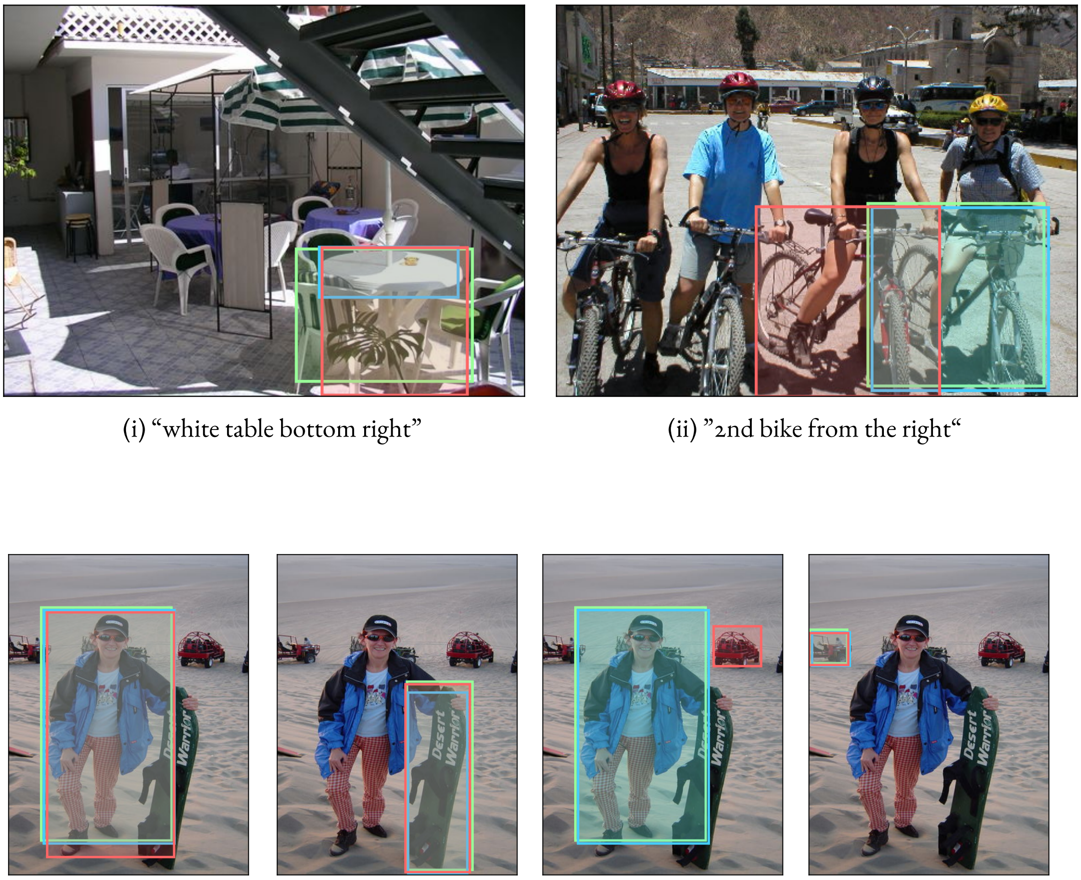
As expected, the *Concept Branch* module, which does not require training, makes the model training more stable and helps to counter the overfitting trend that occurs with small datasets.

6.5 Model Ablation

This section assesses the performance of the model's components: (i) the untrained *Concept Branch*, (ii) the trained visual and textual branches, (iii) and the *Relative Positional Information* component.

The model achieves the best results when both the *Concept Branch* and the trained modules jointly work to produce the final predictions, as shown in Table 6.2. The boost in *Accuracy* given by the *Concept Branch* is significant: $+38.58\%$ and $+30.41\%$ for Flickr30k Entities and ReferIt, respectively. As expected, the *Relative Positional Information* component constantly improves the model accuracy by $+0.1\%$ on Flickr30k Entities and by $+2.6\%$ on ReferIt. Further investigations showed that Flickr30k presents few spatial references in the queries, which explains the difference in performance gains between the two datasets.

6.6 Qualitative Examples



(i) "white table bottom right"

(ii) "2nd bike from the right"

(iii) "a woman"

(iv) "snowboard"

(v) "vehicle to the right of woman"

(vi) "farthest left dune buggy"

Figure 6.4: Examples of predictions obtained with the presented approach. It is delimited in **red** the ground truth bounding boxes, in **green** the final model prediction, and in **light blue** the best bounding box according to the *Concept Branch*.

In this section, in Figure 6.4, it is presented some qualitative examples predicted by the proposed approach. It is highlighted in **red** the ground truth bounding boxes, in **green** the final model prediction, and in **light blue** the best bounding box according to the *Concept Branch*.

In the image 6.4.(i), the concept branch is doing a good job identifying the table location following spacial relation constraints. The trained network then adjusts the rough prediction selecting a bounding box that best encompasses the table. It's worth noting that without the trained visual and textual branches, this example would be misaligned. The image 6.4.(ii) shows a limitation of the proposed approach: among several bike candidates, it follows the prediction of the *Concept Branch* through spatial information and selects the rightmost one, although, it is not correct. This happens because the *Relative Positional Information* module does not consider modifiers like "2nd" to the position "right", and thus the model is guided to the wrong alignment. Images 6.4.(iii)-(vi) depicts the model's efficacy in working on both foreground and background objects, but also its limited ability to understand the context in natural language queries. The example 6.4.(v) asks for "vehicle to the right of woman", but both the *Concept Branch* and the trained modules predict the woman. The misalignment is due to the score returned by the *Concept Branch* to the head "woman" with the bounding box classified as "woman", which is higher than the score obtained with the head "vehicle" and the bounding box "truck".

6.7 Related Works

The proposed approach is related mainly to the weakly-supervised visual-textual grounding area of research. More details about the weakly-supervised Visual-Textual Grounding State-of-the-Art are presented in Section 3.2. This is the first approach based on the principle of first predicting a rough alignment among phrases and boxes adopting a module that does not require training, and then refining those alignments using a learnable neural network.

6.8 Discussion About Large Language Models

The design of the presented approach is well-suited for GloVe, Bottom-Up Attention, and LSTM components. However, these approaches are no more State-of-the-Art. Modern approaches, such as Large Language Models (LLMs) like BERT [43], could improve the performance of the proposed model. Indeed, LLMs take advantage of their effective contextual capabilities to embed words in a sentence. In the proposed architecture, LLMs can replace:

(i) the LSTM in the *Textual Branch*, and (ii) the current GloVe embeddings in the *Concept Branch*. In both cases, the introduction of this new component is not straightforward, especially in the *Concept Branch*. In fact, the concept similarity scores are computed between the head of the phrase and bounding box classes. Thus, it is not clear what context the LLMs should consider during the embedding of class labels.

6.9 Conclusion and Future Work

This chapter introduced the second contribution of this Ph.D. thesis regarding the resolution of the traditional Visual-Textual Grounding task. More in detail, this chapter focuses on solving the weakly-supervised setting. It proposed a model based on the principle of first predicting a rough alignment among phrases and boxes adopting a module that does not require training, and then refining those alignments using a learnable neural network. The model is trained to maximize the multimodal similarity between an image and a sentence describing that image while minimizing the multimodal similarity of the same sentence and a new unrelated image, carefully selected so to help as much as possible during training. In light of this, the proposed model achieved a State-of-the-Art performance on two well-established datasets: the Flickr30k Entities and the ReferIt datasets. Moreover, thanks to the untrained component, the model can be trained just with a small fraction of training examples without deterioration in results.

Future work aims to incorporate an attribute detection module in both noun phrases and bounding boxes, enabling the model to discriminate bounding boxes in a better way, thus suggesting better phrase-boxes alignments. In addition to that, inspired by [121], future works aim to extend the loss function to include a bounding box regression component, that has been proven to help in achieving better accuracy values. Finally, inspired by [15], future works aim to incorporate knowledge graph information in the model, enhancing the *Concept Branch* module with more structured information.

7

Potential Extensions of the Presented Approaches

This chapter proposes two potential extensions that can be adopted for solving visual-textual grounding. The first proposal, which is presented in Section 7.1, regards the adoption of a new set of class labels to adopt when using the Bottom-Up Faster R-CNN [28] (BUA) object detector for detecting objects in the two-stage visual-textual grounding task. This object detector is also adopted by the approaches presented in Chapter 5 and Chapter 6. The new set of classes aims to reduce the noise in the original labels to improve the BUA's detection capabilities and thus, to improve the detection of the visual objects in the visual-textual grounding.

The second proposal, which is resented in Section 7.2, is related to the probability function $\mathbb{P}\left(X_{\boldsymbol{b},V} \mid X_{\boldsymbol{q},V}, \boldsymbol{Z_b}, \widehat{\boldsymbol{Z_b}}, KG\right)$ introduced in Section 4.4. To estimate this function, a solution could be that of adopting an object detector that can use the information conveyed through the set of variables $X_{\boldsymbol{q},V}$ and the graph KG to locate and classify the objects depicted in the image. However, a thorough search of the relevant literature yielded that such object detector has not yet been explored. For this reason, Section 7.2 proposes a new object detector that exploits the graph information and that can be conditioned to search only the objects related to a set of nodes (i.e., $X_{\boldsymbol{q},V}$) of interest. More in detail, it presents a method to condition an existing object detector with the user's intent, encoded as one or more concepts from the WordNet graph, to find just those objects of interest. Albeit according to the probability function the whole graph KG can be observed, in the presented approach the observation is restricted to only the subset of the graph which comprises the nodes related by parent-child relation.

7.1 Cleaner Categories Improve Object Detection and Visual-Textual Grounding

Object detection is the task of locating and classifying the objects depicted in an image [125]. This is a core task in the computer vision field that is used whenever there is the need to localize and recognize objects in images, such as when an autonomous driving car needs to recognize road signs, people, and objects in the streets.

Object detectors are a very important component for solving the visual-textual grounding problem. However, more in general, object detectors are the cornerstone of multimodal vision and language (V&L) tasks, which require joint reasoning over visual and linguistic input.

The object detector should be able to identify many different objects and classify them correctly. Nevertheless, the increase in the number of objects to be recognized usually leads to a more challenging classification problem. The importance of the correct classification of an object is even greater when considering the graph in the resolution of the visual-textual grounding task. In fact, the semantic information conveyed through the classes is crucial to identify the nodes of the graph that best characterize the objects depicted in the image. Section 4.3.5 and Section 4.4 present an explicit example of the existing relation between object detector classes and knowledge graph nodes, where each bounding box is associated with a unique node v_b in the knowledge graph according to its class.

The Bottom-Up Faster R-CNN [66] (BUA) object detector is one of the most commonly-used black box object detectors in the field. Within the V&L literature, it is the defacto standard feature extractor used to represent the visual input [126]. This object detector is also the one adopted by the approaches presented in Chapter 5 and Chapter 6. BUA is pre-trained on the Visual Genome dataset [32] to detect 1600 objects, e.g., “chair”, ”horse”, “woman”, and also to predict their attributes, e.g., “wooden”, “brown”, “tall”. Both the category and attribute set are derived from the freely annotated region descriptions in the Visual Genome dataset, rather than using pre-defined categories like in ImageNet [127] or COCO [128]. Anderson *et al.* [66] did attempt to filter the categories and attributes to prevent near-duplicates, however, the resulting 1600 categories are still imperfect. There are synonymous categories ("wrist watch", "wristwatch"), categories representing single and plurals of the same concepts ("apple", "apples"), ambiguous, difficult to differentiate, categories ("trousers", "slacks", "chinos", "lift"), and categories that actually represent attributes such as "yellow" or "black". Having to predict these noisy categories is likely to prevent the object detector from supporting downstream tasks well.

This section[1] proposed a new set of categories that can be used to train the BUA object detector on the Visual Genome dataset. The new set is the result of a cleaning process performed manually by a native English speaker. Starting from the original 1600 noisy categories, the ambiguous categories were merged to build the final set of 878 clean categories. Then, these clean categories are used to re-train the BUA object detector. In addition to evaluating its object detection performance, the model's feature embedding space and the benefits of using its features in a downstream referring expression comprehension grounding task are analyzed. In the performed experiments, the BUA model trained with the cleaned categories detects objects better, and, examining its feature space representation, it learns a better-clustered embedding space than the model trained with the original noisy categories. The new embedding space produces better bounding boxes feature representations, which in turn can improve performance on a downstream visual-textual grounding task.

The contributions of this section are summarized as follows:

1. starting from the 1600 noisy categories developed by [66], it is proposed a cleaner set of 878 categories with less noise and fewer near-duplicates;

2. it is shown that a BUA detector trained on these cleaned categories improves object detection performance and produces a better visual embedding space compared to using the original noisy categories;

3. finally, it is shown that using the new detector as a black-box feature extractor can improve performance on a downstream visual-textual grounding task.

7.1.1 Recap: Bottom-Up Faster R-CNN

The Bottom-Up [66] model is based on the Faster R-CNN [28] object detector devised to recognize instances of objects belonging to a fixed set of pre-defined categories and localize them with bounding boxes. Faster R-CNN initially uses a vision backbone, such as

[1] Part of this work is published in [129].

ResNet [72] or a VGGNet [73], to extract image features from the image. Then Faster R-CNN applies a Region Proposal Network (RPN) over the input image, that predicts a set of class-agnostic bounding box proposals for each position in the image. The RPN aims to detect all the bounding boxes that contain an object, regardless of what the object is. Then, for each detected bounding box proposal, Faster R-CNN predicts a class-aware probability score and a refinement of the bounding box coordinates to better delimit the classified object. The Faster R-CNN multi-task loss function contains four components, defined over the classification and bounding box regression outputs for the Region Proposal Network and the final bounding boxes refinement.

The BUA object detector initializes its Faster R-CNN backbone weights from a ResNet-101 [72] model pre-trained on the ImageNet [127] dataset for solving the image classification task. The model is trained on the Visual Genome [32] dataset to predict 1600 different objects. Since the Visual Genome dataset also annotates a set of attributes for each bounding box in addition to the category it belongs to, the BUA model adds an additional trainable module for predicting attributes (in addition to object categories) associated with each object localized in the image. For this reason, the BUA model adds a multi-class loss component to the original Faster R-CNN losses to train the attribute predictor module.

The 1600 categories used to train the BUA model were set by [66]. The Visual Genome dataset annotations consist of image regions associated with region descriptions (natural language strings) and the attributes of the object depicted in it. [66] extract category labels from the region descriptions, but their procedure is underspecified (for example, it is unclear if they used a part-of-speech tagger to extract nouns and adjectives as labels for objects and attributes). They filtered the original set of 2500 object strings and 1000 attribute strings based on object detection performance, resulting in a set of 1600 categories and a set of 400 attributes. However, the remaining set of categories is still noisy. It contains plurals and singular of the same concepts, such as "dog" and "dogs", overlapping categories such as "animal", "cat", and "dog". Moreover, it contains near-duplicate categories such as "motorcycle" and "motorbike", unhelpful distinctions like "lady" and "woman", labels representing attributes such as "yellow" and abstract notions like "front". These noisy labels may result in a sub-optimal representational space and likely impair the ability of the model to classify objects correctly. Given that several labels equivalently express the same meaning, whenever the model needs to predict a category for an object appearing in the image, the model needs to split its predicted probabilities among all equivalent categories. This probability split occurs not only when two or more categories express the same meaning (e.g., "hamburger" and "burger") but also when the meanings expressed by the categories overlap substantially, such as the categories "pants", "trousers", and "slacks".

7.1.2 Cleaning the Visual Genome Category Set

This section proposes a new set of categories to use for training the BUA object detector. This new label set is the outcome of a cleaning process applied to the 1600 original categories by the authors of this work, which include native English speakers. This process aimed to combine ambiguous and low-frequency categories together. During the cleaning process, the categories were joined together according to the following principles:

1. **Plurals**: singular and plurals categories, such as "giraffe" and "giraffes". In most instances, these annotations represent the same concept and should be treated as the singular category. This led to 258 category merges.

2. **Tokenization**: categories with and without spaces, such as "wrist watch" and "wristwatch", should be treated as the same category. This resulted in 29 category merges.

3. **Synonyms**, such as "microwave" and "microwave oven", "hamburger" and "burger", express similar concepts with minor differences that are usually not important. Often, as in "microwave oven", these are compound phrases that can be identified automatically, though it is important to verify them manually (e.g., "surf" and "surf board" should not be merged).

4. **Over-specific** categories with substantial annotator disagreement where several words are used interchangeably, e.g., "pants", "trouser", "sweatpants", "jean", "jeans", and "slacks".

However, during the cleaning process, it was not always clear when to merge the categories since: (i) some categories are inherently ambiguous, such as "home". (ii) some categories are abstract and don't have the meaning of a concrete object, such as "items", "front", "distance", "day". (iii) some categories represent attributes rather than objects, such as "yellow" and "black"

For some ambiguous labels like 'lot' or 'lift', visual inspection of the labeled images showed that within VG, these labels were used mostly to refer to one concept: "lot" usually showed car parking and was merged with "parking lot", similarly "lift" was merged with "ski lift". In other cases, no single meaning predominated and these labels were left unmerged (e.g., 'stand' was not merged with either 'baseball stand' nor 'tv stand'). The abstract and attribute categories were also left as they were. In this way, the adopted cleaning process defines a surjective function that maps the original labels set to cleaner labels set.

The cleaning process produces a new set of 878 categories[2] from the original 1600 categories (Appendix D.1). Figure 7.1 shows frequencies of objects appearing in the Visual Genome training split, where objects are either labeled according to the original label set (in blue) or the new cleaned label set (in orange). The new labels lead mostly to the removal of many low-frequency categories in the long tail, rather than creating new very frequent categories.

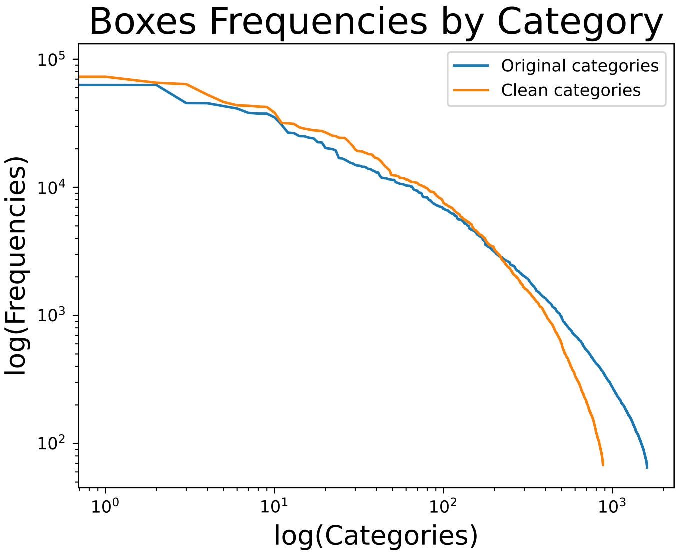


Figure 7.1: LogLog plots of objects frequencies for each category. The frequencies are calculated on the training set annotations. The distribution of the original categories is in blue, and the new categories are in orange. The cleaning process did not generate high-frequency categories and at the same time removed many low-frequency categories.

7.1.3 Experimental Setup

A BUA object detector is trained matching the procedure of Anderson *et al.* [66], except that the new clean categories are used as object labels instead of the original noisy categories.

[2] https://github.com/drigoni/bottom-up-attention.pytorch/blob/master/evaluation/objects_vocab.txt

7.1.3.1 Datasets and Evaluation Metrics

Following [66], the training and test data for the models is the Visual Genome (VG) [32] dataset. It is a multipurpose dataset that contains annotations of images in the form of scene graphs that form fine-grained descriptions of the image contents. It supplies a set of bounding boxes appearing in the image, with labels such as objects and persons, together with their attributes, such as color and appearance, and the relations between them. The original VG labels were converted to object labels by [66], as described in Section 7.1.1. Note that the BUA model used in this work is trained only using the VG training split, unlike some pre-trained models available, e.g., in the MILVLG repository, which uses both training and validation splits for training.

To assess the object detectors' performance, this work uses the Mean Average Precision (AP) metric, which is the standard metric for measuring the accuracy of object detectors such as Faster R-CNN [28]. All evaluation results presented in this section are obtained on the VG test split. Average precision uses an Intersection over Union (IoU) threshold of 0.5 to determine whether the predicted bounding box is sufficiently similar to the gold region. See Appendix A.3 for more details about the IoU metric. The AP metric is distinguished between 'macro' and 'micro' (also known as 'weighted') AP: MacroAP weights each category uniformly (macro-averaging class-wise precision) while MicroAP weights each category by the number of items in the category (equivalent to micro-averaging over all items, regardless of class). MacroAP will emphasize the effect of small categories, while MicroAP will be dominated by the most frequent categories.

Precision is indirectly affected by the number of categories in the label set: e.g., a random baseline over 100 categories will perform worse than a baseline over 10 categories. Since the objective of this section is to compare models with different numbers of categories, this is an unavoidable confound. To mitigate against it, for the original model, which predicts labels in the original label set, predictions are mapped to the clean label set. For example, if the model predicts 'motorcycle' in the original label set, this prediction gets mapped to the same category ID as the model's 'motorbike' predictions, because these two labels have been collapsed in the clean label set. This results in mapped predictions with the same number of categories as the clean label set predictions, which means that comparison between label sets is fairer. However, this procedure also removes all errors due to confusing the two labels that have been merged in clean (e.g., if the original gold label for the 'motorcycle' prediction was 'motorbike', this incorrect prediction is now counted as correct), which makes it a very strict evaluation.

7.1.3.2 Random Baseline

The BUAdetector trained on the cleaned classes is also compared against a BUA detector trained with a randomly merged category set. The randomly merged set was created by randomly selecting pair of categories in the original set to combine until it reached the same number of categories adopted in the clean set (i.e., 878). This procedure leads to a distribution of category sizes that is very similar to the clean label set, see Appendix D.1. However, the randomly merged categories will include semantically very distinct objects, e.g., bananas and motorcycles are in the same category. This allows one to separate the effect of having cleaner categories from the effect of simply having fewer categories.

7.1.3.3 Implementation Details

For the development of this work, the code available in the MILVLG[3] repository was used, which is a Pytorch implementation of the original Caffe[4] model. In particular, the MILVLG code allows to train, evaluate, and extract bounding boxes from images using both the Detectron2 framework[5] as well as the original Caffe model weights. When not explicitly indicated, it is used BUA implemented with Detectron2. Between 10 and 100 bounding boxes are extracted for each image in input. The default MILVG hyper-parameters were used, apart from setting the batch size to 8, and training only on the training data split. when training on the new label set the same default hyper-parameters from the model trained on the original 1600 categories are used. The object detectors are trained for 180K iterations. All experiments were performed in a distributed parallel system using a V100 32GB GPU.[6]

7.1.4 Experiments

The experiments compare BUA models trained on the new smaller label set with the original BUA model using the original label set. These two models are compared in terms of performance on the original object detection task, the properties of the embedding space learned by the detector, and the utility of the features in a visual-textual grounding task on

[3] https://github.com/MILVLG/bottom-up-attention.pytorch

[4] https://github.com/peteanderson80/bottom-up-attention

[5] https://github.com/facebookresearch/detectron2

[6] https://github.com/drigoni/bottom-up-attention.pytorch

Model	Implementation	Visual Genome (%)	
		MacroAP50↑	MicroAP50↑
BUA Original	Caffe	9.37	15.14
BUA Original	PyTorch	9.10	15.93
BUA Original→Clean-878	PyTorch	10.72	17.34
BUA Clean	PyTorch	11.01	17.60
BUA Original→Random-878	PyTorch	9.49	15.79
BUA Random	PyTorch	9.46	15.61

Table 7.1: BUA object detection results on the Visual Genome dataset. The model trained on the clean categories,“ BUA Clean”, achieves better object detection performance than the model trained on the original categories. “BUA Original→Clean-878” and “BUA Original→Random-878” are results from models trained on the originalcategories whose predictions are mapped to clean and random label set respectively, to match label set size (878 labels in both cases)

the Flickr30k Entities dataset. It is expected the removal of label ambiguity in the new label set to lead to better performance on object detection and visual-textual grounding.

7.1.4.1 Object Detection

The object detection is tested on the Visual Genome test set: see Table 7.1. The model trained on the new labels, BUA Clean, outperforms the BUA Original model by nearly two points on macro and micro AP.

To check how much of this improvement is due to simply having a smaller label set, BUA Clean and BUA Original are compared against the random (i.e., BUA Random) baseline (where categories were iteratively merged to the same number of labels as the clean set) and against the same original predictions, but with predicted labels mapped to the clean set (e.g., predictions for ‘egg’ and ‘eggs’ are mapped to the same label, as in the clean set). The BUA Random results are slightly worse than the BUA Original model, indicating that fewer labels on their own are not enough to micro or macro AP. Mapping the original predictions to the new labels improves both metrics, indicating that many of the mistakes in the BUA Original model are due to confusion between labels that are merged in the clean set. However, performance does not reach the level of BUA Clean model, demonstrating that using better labels at training time is important. Since this improvement is visible in both micro and macro AP, the new labels do not only improve frequent categories (reflected in MicroAP) or infrequent categories (MacroAP).

Figure 7.2 shows how noise in the category set affects the prediction confidence of the model. By ‘prediction confidence’, it is meant the probability assigned to the argmax cate-

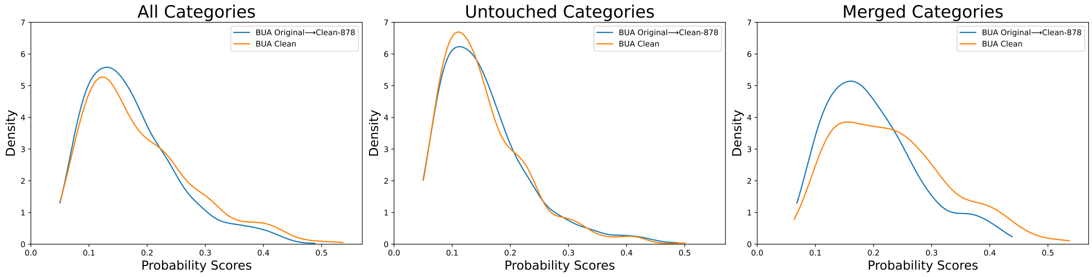


Figure 7.2: KDE plots for the probability values of the argmax category predicted by the model. The plots on the left consider all the categories, the plots in the center consider just the categories that were not merged during the cleanup process (i.e., "Untouched"), and the last plots on the right consider only the merged categories. Overall, the cleaned categories lead to higher confidence values than the original categories.

gory predicted by the model when it detects an object. These maximum probability detections play an important role in determining which detections to use in downstream tasks.[7] Results show that the BUA detector trained on the cleaned categories produces more high confidence predictions than a detector trained on the original noisy categories. Closer inspection shows that this difference is due to higher confidence when predicting objects in the new merged clean categories. This confirms the hypothesis that the original categories result in probability mass being split across multiple synonymous labels, and this issue is resolved by the new cleaned categories. The same behavior is not visible with random categories.(Appendix D.2).

These results support the hypothesis that noise and repetition in the original label set make it difficult to learn good distinguishing features between categories. They also imply that it is necessary to retrain the object detector on cleaner labels to fully improve its detection capabilities on downstream tasks.

The experiments also show differences in the performance of the BUA Original model as implemented in Caffe and Pytorch, despite the fact that Pytorch is meant to be a reimplementation of the Caffe version. Similar behavior will be visible in the visual-textual grounding experiment later on, where the difference between the two implementations is more substantial.

7.1.4.2 Feature Space Analysis

This section attempts to characterize the differences in feature space, given features from a model trained with the clean label (i.e., Clean) set vs the original model (i.e., Original). The

[7] In V&L pretraining, it is common to use the (10-100) most confident regions [126] detected in each image.

K	Th.	All Neighbors (%)			Filtered Neighbors (%)		
		Original	Random	Clean	Original	Random	Clean
1	0.05	12.15±12.25	12.36±11.15	17.30±14.79	37.32±15.07	37.83±12.32	42.34±15.82
5	0.05	24.33±13.38	24.91±12.01	29.74±15.10	34.16±13.78	34.68±12.24	39.09±15.09
10	0.05	27.76±13.23	28.37±11.87	32.96±14.85	32.91±13.71	33.48±12.19	37.84±15.11
1	0.2	51.02±22.74	51.88±20.91	55.36±20.03	69.22±18.99	70.03±16.76	71.96±17.54
5	0.2	60.40±19.75	61.47±17.84	63.92±19.00	65.12±19.68	66.12±17.54	68.29±18.58
10	0.2	60.55±20.18	61.71±18.20	64.16±19.31	62.95±20.43	64.05±18.34	66.32±19.39

Table 7.2: Proportion of K-nearest neighbors that share the same predicted category. Results were obtained with the models trained on the original, the random, and the clean categories. Overall, at each value of K, the embedding space of the model trained on clean categories is better clustered than those of models trained on the original and random labels.

features are from the ResNet-101's `pool5_flat` layer; these are the most common representation used for downstream tasks (e.g., visual-textual grounding). For each image in the VG validation set, the features corresponding to the bounding box proposals are extracted. Two confidence thresholds are tested: with th=0.05, the models return approximately 280000 bounding box feature vectors, whereas, with th=0.2, it only evaluates approximately 100000 features. (Different models return slightly different but comparable numbers of proposals.)

In order to be useful for downstream tasks, it is expected that bounding boxes that contain similar objects should have similar features and the same predicted categories. This is tested using nearest neighbors and cluster analyses.

Nearest Neighbors The local structure of the feature space can be examined using a nearest neighbors analysis: for each point in the embedding space (i.e., bounding box features), it is calculated the proportion of K (with $K = 1$, 5, and 10) nearest neighbors that share the same category. This analysis is not affected by the different number of labels in the several sets and therefore it allows one to fairly compare models' embedding spaces. It is expected the embedding space of the model trained with cleaner categories to be clustered better than the other embedding spaces. In other words, it is expected that each point has more neighbors that share the same category when using cleaned labels.

Table 7.2 reports the results of this analysis, considering features extracted with different threshold values (i.e., 0.05 and 0.2) and considering either all features or only features from different images ("Filtered Neighbors"). This step removes features that might be from highly overlapping regions of the same image.

Overall, as expected, the bounding boxes extracted by the model trained on the cleaned label set have higher proportions of nearest neighbors that share the same category. This

Th.	K	Categories	All Neighbors (%)		Filtered Neighbors (%)	
			Original	Clean	Original	Clean
0.05	1	All	12.15±12.25	17.30±14.79	37.32±15.07	42.34±15.82
0.05	1	Untouched	9.19±9.47	8.56±8.84	32.20±16.21	32.86±15.18
0.05	1	Merged	12.71±12.62	19.03±15.12	38.28±14.65	44.22±15.26
0.05	5	All	24.33±13.38	29.74±15.10	34.16±13.78	39.09±15.09
0.05	5	Untouched	19.71±12.27	20.35±11.77	28.62±24.39	29.48±13.68
0.05	5	Merged	25.19±13.40	31.60±14.99	35.19±13.41	40.99±14.63
0.05	10	All	27.76±13.23	32.96±14.85	32.91±13.71	37.84±15.11
0.05	10	Untouched	22.55±12.64	23.33±12.26	26.97±14.15	27.95±13.58
0.05	10	Merged	28.73±13.12	34.87±14.57	34.01±13.34	39.80±14.62
0.2	1	All	51.02±22.74	55.36±22.03	69.22±18.99	71.96±17.54
0.2	1	Untouched	43.34±21.95	41.37±21.45	62.26±23.11	60.92±22.20
0.2	1	Merged	52.14±22.64	57.29±21.40	70.23±18.09	73.48±16.22
0.2	5	All	60.40±19.75	63.92±19.00	65.12±19.68	68.29±18.58
0.2	5	Untouched	51.88±21.68	50.58±20.88	56.33±23.38	55.51±22.08
0.2	5	Merged	61.64±19.14	65.75±17.97	66.40±18.74	70.05±17.32
0.2	10	All	60.55±20.18	64.16±19.31	62.95±20.43	66.32±19.39
0.2	10	Untouched	50.83±22.63	49.92±21.42	52.89±23.80	52.12±22.38
0.2	10	Merged	61.97±19.39	66.12±18.15	64.42±19.46	68.28±18.09

Table 7.3: Proportion of K-nearest neighbors that share the same predicted category, comparing models trained using the original versus the clean categories. (See Table D.1 for a comparison with random categories.) "Th." indicates the threshold values adopted for bounding box extraction. "Merged" refers to original categories that are merged into one new clean category. "Untouched" refers to those categories not merged with others during the cleaning process, and "All" refers to all the categories. Overall, the clean features are better clustered than the original features.

difference is substantial and consistent across different values of K, thresholds. Table 7.3 shows that the improvement is due to better neighborhoods of features with merged labels, and only in some cases better features of unmerged, original labels.

The random features (i.e., Random) present results very similar to those obtained with the Original features, but with a small improvement. Surprisingly, this improvement is most evident for features of categories that are the same between Original and Random (Appendix D.3), rather than the categories that were merged in Random, suggesting that there is an advantage to training on fewer labels overall.

Surprisingly, when features from the same image are ignored (Filtered Neighbors), the percentage of neighbors who share the same category increases dramatically. This indicates that BUA features tend to place visually similar regions (from the same image) close together, regardless of their semantic content (their predicted object label).

In conclusion, the analysis of the neighbors verified the main claim: when the BUA object detector is trained with the original noisy labels, it results in a sub-optimal representational space that can be improved simply by retraining the model on cleaner labels set.

Distances This section examines the global structure of the feature space by looking at the distances between items with the same label (intra-category) and the distances between the category centroids (inter-category). The hypothesis is that if the feature space is organized by categories, then intra-category distances should be small, while inter-category distances should be larger.

Table 7.4 reports the inter and intra-category distances for features from the models trained with the original, clean, and random labels. Intra-category distance is the average Euclidean distance between features with the same predicted label, while inter-category distance is the average Euclidean distance between the centroids of each category (all averages are macro-averages over categories). Results show that the Clean labels lead to categories that are clustered more closely together, evident in a lower average intra-category distance, compared to both the Original and Random labels. Counter to the hypothesis made at the beginning of this section, inter-category distance is lower when using Clean labels, especially compared to the Original labels, and also slightly lower than Random labels. This indicates that the global feature space is also more compact overall. Surprisingly, across all feature spaces (Original, Clean, and Random) the intra-category distances are higher than the inter-category distances, suggesting that features from different categories are highly intermingled.

In order to control for label set and category size, the original features are mapped to the clean (i.e., "Orig.→Clean-878") or random (i.e., "Orig.→Random-878") set of categories, ensuring the same number of points in each label category, as well as the same number of labels. This results in a higher intra-category average distance, compared to the original categories, which indicates that features from merged labels are not mapped to nearby parts of the space. Notably, the clean mapping leads to only very slightly lower intra-category distances compared to the random mapping.

Overall, the analysis of the local neighborhoods shows a positive effect of the clean label set, with more neighbors with the same label. However, the analysis of the global feature space suggests that the BUA features are not well separated according to object semantics, regardless of the label set used.

Analysis	Orig.	Orig.→Clean-878	Clean	Orig.→Random-878	Random
Intra-Category	49.69 ±8.64	52.10 ±8.10	45.37 ±6.98	52.96 ±8.63	47.77 ±7.87
Inter-Category	47.97 ±5.31	NA	39.76 ±4.94	NA	40.19 ±5.87

Table 7.4: Intra-category (average pairwise of points with the same label) and inter-category (average distance between categegory/label centroid) Euclidean distances in different feature spaces. Results were obtained with the models trained on original (i.e Orig.), clean, and random label sets. The model trained on cleaner labels presents lower distances in both the intra-categories and the inter-categories analysis.

7.1.4.3 Visual-Textual Grounding Results

This section investigates the utility of the features extracted with the BUA model in the visual-textual grounding task on the Flickr30k Entities dataset. See Appendix A.1 for more details about the Flickr30k Entities dataset. The expectation is that features extracted with the models trained on the new categories will be more coherent and useful than those extracted with the model trained on the original set of categories, leading to better performance on this downstream task.

In this section, the Bilinear Attention Network (BAN) [4] model is used as the visual-textual grounding model, which, even if no longer State-of-the-Art, obtains relatively good results on the Flickr30k Entities dataset. The advantage of using the BAN model is that it is a simple model that uses a straightforward fusion component to merge the text and visual information, and that requires only the Flickr30k Entities dataset for training (other models that achieve higher scores are pre-trained on much larger data sets and have more complex architecture [11, 130, 131, 132, 133]). BAN implements a simple architecture[8] that uses only the 2048-dimensional bounding box features extracted from the object detector as the visual input features; it does not use the label predicted from the features. On the text side, the model initializes each word with its GloVe [3] embedding and uses a GRU [134] to generate a representation for the sentence. The visual and textual representations are then fused together through a bilinear attention networks. The simple fusion component allows one to see the effect of different visual feature spaces more clearly. The code provided by the authors is used[9], and no hyper-parameters were changed from the original model. The experiments were performed using an A5000 24GB GPU.

Table 7.5 reports the results obtained in the visual-textual grounding task by the BAN

[8] The model is composed of about 19M trainable neurons.

[9] https://github.com/jnhwkim/ban-vqa

Features	Threshold	Test Set (%)				N. Bounding Boxes		
		R@1 ↑	R@5 ↑	R@10 ↑	UB↑	Min	Max	Test
BAN [4]	0.2	69.80	84.22	86.35	87.45	10	91	30 034
Original	0.2	73.32	84.21	85.67	86.53	2	89	20 916
Clean	0.2	73.41	85.08	86.52	87.31	2	93	21 923
Original	0.1	74.72	86.06	88.71	90.70	5	100	36 792
Clean	0.1	75.43	86.76	89.56	91.22	7	100	36 719
Original	0.05	75.41	85.46	88.86	92.38	12	100	59 256
Clean	0.05	75.75	85.88	89.52	92.67	11	100	56 731

Table 7.5: Visual Grounding results obtained with the Bilinear Attention Networks (BAN) [4] model on the Flickr30k Entities dataset. "R@K" refers to the Recall metric with the top K predictions, while "UB" refers to the upper bound results that can be achieved with the bounding boxes extracted with the indicated threshold. The features extracted with the model trained on the clean labels set consistently perform better than the original features.

model trained using the features extracted by both the models trained on the original (i.e., Original) and new cleaner (i.e., Clean) label sets. Whenever BAN is trained using the Clean features, the performance of the model increases compared to the BAN model trained on the Original features. The improvement is small but consistent across bounding box thresholds and recall levels.

The results also show that the BUA PyTorch implementation of the BAN model always achieves better performance than the Caffe implementation, even with fewer bounding boxes. This result implies that the implementation code used to train the object detector strongly impacts the results of the visual-textual grounding task, although, in the object detection task, there is only a small improvement[10].

In conclusion, the results obtained with the BAN model on the visual-textual grounding task suggest that the BUA model trained using a cleaner set of labels presents not only a well-clustered embedding space but also a more useful features representations able to improve downstream tasks.

7.1.5 Related Work

The work proposed in this section relates to (i) work that adopts the Bottom-Up model [66] for the detection of objects depicted in images, especially for multimodal downstream tasks,

[10] The extracted features used in the BAN paper are not made available by the authors. However, some 'reproducibility' features (slightly different) were made available by third users (`https://github.com/jnhwkim/ban-vqa/issues/44`) who successfully reproduced the main paper results.

and (ii) work that addresses learning neural networks with noisy labels. More details regarding the Bottom-Up model are presented in Section 7.1.1.

7.1.5.1 Bottom-Up for Object Detection

Many object detectors exist, that differ according to their ability to detect objects in the image, the computing power required for their use, and their ability to recognize a large set of different objects [66, 67]. An object detector should be able to identify many different objects [23] and classify them correctly. The appeal of BUA features lies in part in the large number of object categories. Nevertheless, the increase in the number of objects to be recognized leads to a more challenging classification problem.

Starting with [66], in which the extracted object detector bounding boxes were used as input to a Visual Question Answering (VQA) model, much work on VQA adopted the BUA model as object detector [135, 136, 137, 138, 139, 140, 141, 142]. BUA features have also been used for the Visual-Textual Grounding task [23, 10, 4, 88, 123, 4]. In addition, many recent large pre-trained Vision and Language models use BUA features as their visual representations [143, 144, 145, 146, 147, 148, 149]. These models are used as the starting point for a wide variety of multimodal tasks, including image description, VQA, natural language visual reasoning, visual-textual grounding, etc [150, 151, 126].

All these works directly depend on the quality of the objects detected by the BUA model. Incorrect identification and/or classification of objects may have major repercussions in the resolution of downstream tasks, making it important to analyze in more detail the labels used to train the BUA model.

7.1.5.2 Noisy Label Sets

This work, aiming to improve data quality by improving label quality, is related to the branch of research area addressing noisy label effects during neural network training. However, most of the work presented in this section addresses the problem of badly labeled data, i.e., noise at the instance level (see [152] for a recent survey).

The work presented in this section is interested in the problem of bad or noisy labels, rather than noisy data. [153] show that their framework for estimating noise in data labeling can also identify 'ontological issues' with the labels themselves. Removing duplicate labels during training improves performance on ImageNet classification, in line with the object detection improvement found in this work. [154] identify and correct label issues in ImageNet for better, more robust model evaluation and comparison; removing 'arbitrary' label distinctions ensures models are not rewarded for overfitting to spurious noise.

[155] aims to discover a 'basic level' label set, i.e., the labels corresponding to the human default or basic level categories, by merging labels that are often confused. They find that training an image classifier on these categories can improve downstream image captioning and VQA.

7.1.6 Conclusion and Future Work

This section introduced a new set of 878 category labels to retrain the BUA model, which refines the originally noisy 1600 categories by merging labels that are synonymous or have highly related meanings. The effect of using the cleaner label set in terms of performance on the original object detection task is investigated, showing that the model trained on the new set of labels improves its object detection capabilities. Also, it was analyzed the embedding space in the object detector trained on the cleaned categories and showed that it is better clustered than the embedding space derived from the original categories. Finally, this section evaluated the utility of the new model as a black-box feature extractor for a downstream visual-textual grounding task with the Bilinear Attention Network model. The results show that features from the new object detector can consistently improve the BAN model across commonly used object detection thresholds.

Future work involves studying the effect of using the improved label set on large pre-trained language-and-vision models, such as VILBERT [147] and LXMERT [146]. Since these models use the bounding box category labels predicted by the object detector in their loss function, in addition to using the features as their visual input, removing label noise should benefit these models. Future work also aims to integrate the object detector trained on the new cleaned classes in the models presented in Chapter 5 and Chapter 6.

In this section, the noisy categories are merged using a skilled human annotator, which may have introduced some unwanted human bias or error into the cleaning process. Nevertheless, the proposed approach highlights the advantage of using improved label sets, both for core object detection and downstream multimodal task performance. Future work could generate alternative cleaned categories by merging similar ones, e.g., using a framework similar to Confidence Learning [153].

7.2 Object Search by a Concept-Conditioned Object Detector

The object detection task aims to find all objects of a given set of object categories shown in an image. In many situations, however, a user looks at a picture with the intent of finding objects of one or more types, which are expressed by any noun, and not restricted to a predefined set of categories (see Figure 7.3). This task is called the "Find-That" task.

A practical example is given by an image object extraction task where a user aims to automatically extract from a stream of images all the occurrences of one or more specific objects (entities), e.g., all the cats and dogs contained in the images. Notice that some images may neither contain cats nor dogs, while others may contain both of them or just only cat(s) or dog(s). For this task, the intent of the user is known a priori, although it may range across a large set of possible intents. Because of that, the intent can be used to condition an object detector to obtain a better recognition rate, and thus a better final performance.

The task described above differs from visual-textual grounding textual expressions [6, 7, 8, 9, 10], as visual-textual grounding has the objective of finding a precise object referred by a textual noun phrase, while the "Find-That" task is interested in finding all the objects related to a **set** of given intents/categories. More distinctions are underlined in the *Related Works* Section 7.2.5.

A baseline method to solve the "Find-That" task is by using an object detector that extracts all the objects in the image and then filters the results according to the specified categories. This last step is not trivial as a user can express her/his interest using nouns that are not in the categories supported by the object detector. Therefore, such a baseline method should use a filtering procedure, which does reconcile the noun specified by the user with the set of supported categories. In such a baseline, the object detector is independent of the user's intent, and it may return many undesired object categories.

This section[11] proposes a method to condition an object detector with the user's intent, represented by one or more concepts of WordNet [33], to drive the localization and classification of only desirable objects. See Section 2.5 for more details about the WordNet graph. Hence, there is the need to modify the object detector so that it takes in input also a set of nouns and focuses its attention only on objects of the categories directly or indirectly specified by the nouns. WordNet allows the reconciliation of the user's intent with the set of supported categories, it handles the synonymous and the problem of multiple meanings as-

[11] **D. Rigoni**, L. Serafini, and A. Sperduti, "Object Search by a Concept-Conditioned Object Detector", *Under Peer Review*.

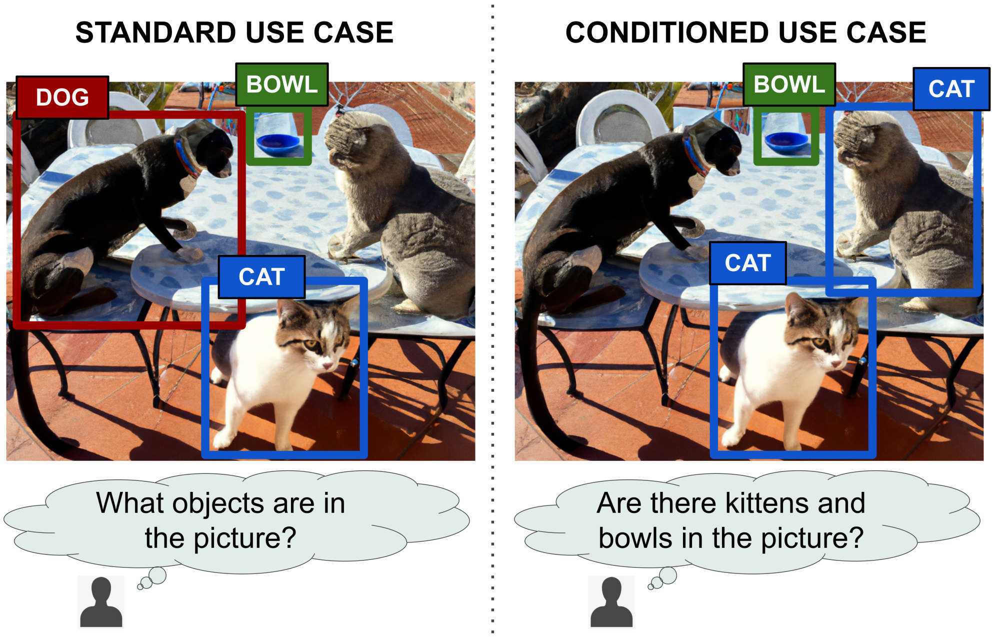

STANDARD USE CASE
DOG
BOWL
CAT
What objects are in the picture?
CONDITIONED USE CASE
BOWL
CAT
CAT
Are there kittens and bowls in the picture?

Figure 7.3: Main differences between the standard object detector and the conditioned object detector approach. (Left) A scenario involving a standard object detector detecting all the objects in the image. (Right) A scenario involving a concept-conditioned object detector, which given the image in input jointly with the user's intent, directly returns only the objects of interest, thus eventually avoiding mistakes, as the missed cat on the top right of the image by the standard object detector.

sociated with the same word (polysemy) that would be present with textual inputs.

Figure 7.4 highlights the main difference between the baseline described above (top), and the proposed approach involving a concept-conditioned object detector (bottom). Starting from the image, a standard object detector detects all the objects depicted in the image and passes them to an ad-hoc post-processing algorithm, which selects only the objects classified with categories that are represented by the WordNet concepts in input. Section 7.2.2 elaborates on how WordNet concepts can be matched with the object detector pre-defined categories, which is an important component for the *Post-processing Selection* component of the model. The proposed concept-conditioned object detector takes in input also a set of concepts, and applies the object detection and filtering phase to a combination of image and *Concept Set Encoding* component. The integration is implemented by the *Fusion Block*, which fuses the visual features returned by the model *Backbone* with the concepts embeddings. Then the multimodal features are used in the *Object Detector Head* to locate and classify all the objects of interest.

With this new approach, however, new datasets are needed to train these models, with

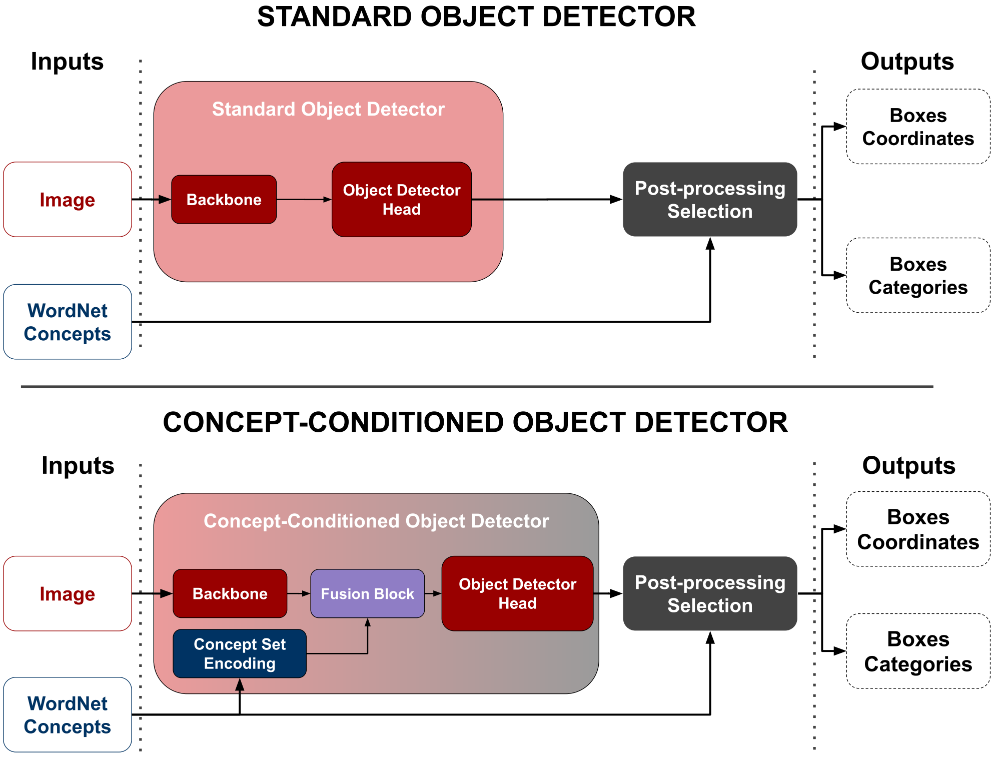


Figure 7.4: Operational setting adopted in this work for finding all the objects contained in an image that represents the user's intent. (Top) A standard object detector, given an image as the only input, detects all the objects which, through the *Post-processing Selection* algorithm, are filtered accordingly to the WordNet concepts. (Bottom) The concept-conditioned object detectors take in input also the WordNet Concepts. The *Concept Set Encoding* encodes in an embedding space the set of WordNet concepts. The *Fusion Block* fuses the visual features returned by the model *Backbone* with the concepts features. Then the multimodal features are used in the *Object Detector Head* to locate and classify all the objects of interest.

inputs made of WordNet concepts, as well as images. In this section, it is proposed an effective strategy to generate WordNet concepts from already existing object detection datasets, removing the need to create new ad-hoc datasets from scratch.

Overall, the contributions of this section can be summarized as follows: (i) it presents a novel approach to focused object search in an image by conditioning existing object detectors with the user's search intent, represented as a set of WordNet concepts. The proposed approach can be implemented with minor changes to a standard object detector software, e.g., it does not require the modification or addition of any object detector loss; (ii) this is the first work that proposes conditioned object detectors in which the user's intent is represented as a **set** of WordNet concepts. The set approach allows the user to search multiple objects at the same time, while the WordNet graph allows the user to express a query using concepts that are not directly associated with the set of pre-defined categories supported by the object detector; (iii) it proposes an effective strategy to generate WordNet concepts from already existing object detection datasets, removing the need to create new ad-hoc datasets from scratch for training concept-conditioned object detectors; (iv) it is empirically shown, on two widely used object detection datasets, COCO and Visual Genome, and several object detection architectures, that the proposed concept-conditioned object detector approach performs better than the standard baseline.

7.2.1 Problem Formulation

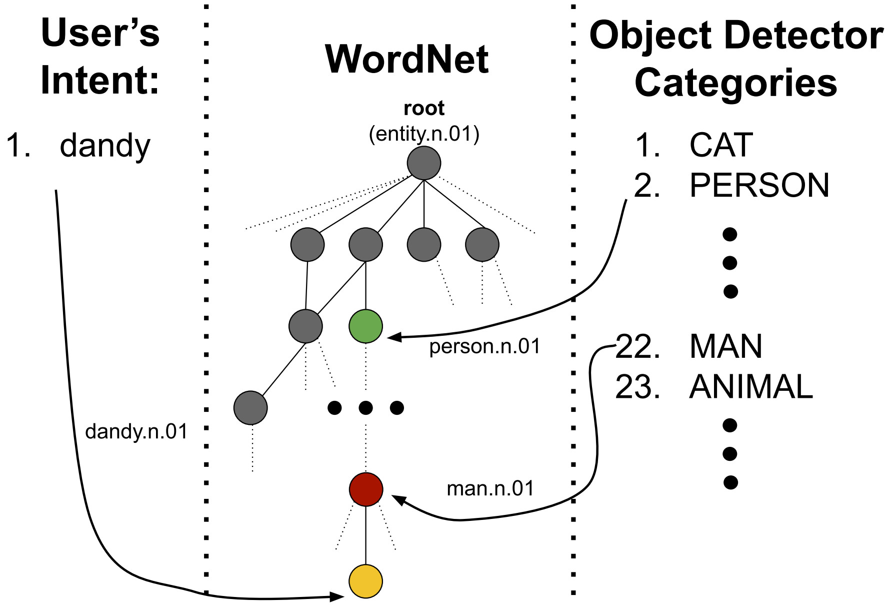


Figure 7.5: A toy example that highlights the difficulties in retrieving the dataset categories given the concepts in input. Given the user's intent "dandy", which refers to the concept colored in yellow in the WordNet graph (dandy.n.01), there are two ancestor concepts, i.e., "man.n.01" and "person.n.01", that are associated with the dataset pre-defined categories "MAN" and "PERSON", respectively.

Before giving a formal definition of the problem, there is the need to clarify an issue about the "intent" of the user, i.e., the expected output from an object detector that takes in input a set of concepts (from WordNet). In fact, given a set of input concepts, it is not straightforward how to retrieve the categories that are represented by those concepts. Although it can be considered safe to assume that any object detector pre-defined category can be mapped to a corresponding concept in WordNet, Figure 7.5 presents a toy example that highlights the main difficulties: (i) WordNet concepts may have multiple concepts as parents; hence, given a concept, the set of all its ancestors potentially could result in a very large set of concepts; (ii) since the object detector's pre-defined categories can be related to each other, e.g., the category "PERSON" is related to the category "MAN", concepts associated with the pre-defined categories can also be related by parent-child relations in the WordNet graph.

Therefore, given a concept, a first approach could be to select all the pre-defined categories whose concepts are equal or ancestors of at least a concept in input. In the example, this means that the selected objects should be classified as "MAN" and "PERSON". However, maybe the user is interested in finding only objects belonging to the "MEN" category and not objects also classified as "PERSON". In that case, the alternative approach would be to select only the category whose WordNet concept is the closest (parent) to the concept in input. In the example, this implies the selection of only the objects classified as "MAN", discarding all the objects classified as "PERSON". In general, one could think of an "intent" that is defined by the intended *concept depth*, i.e., how far to travel the WordNet graph to retrieve the object detector categories. To cope with this issue, in the following, the problem is formally defined by also specifying a *concept depth*.

Let $\mathcal{L}$ be the set of categories supported by an object detector, $\mathcal{S}$ the set of concepts in WordNet, $f : \mathcal{L} \to \mathcal{S}$ a function that associates to each category of the object detector a unique concept in WordNet. For every $d \in \mathbb{N}$ let $f^d : \mathcal{L} \to 2^{\mathcal{S}}$ be the function that maps a label $l \in \mathcal{L}$ of the object detector into the set of WordNet concepts defined as:

$$
\begin{aligned}
f^0(l) &= \{f(l)\}, \\
f^{d+1}(l) &= f^d(l) \cup \left\{ s \in \mathcal{S} \;\middle|\; \begin{array}{l} \exists s' \in f^d(l) \text{ such that } s' \\ \text{is a parent concept of } s \text{ in} \\ \text{WordNet.} \end{array} \right\}.
\end{aligned}
$$

Let $G(\boldsymbol{I}) = \{(\boldsymbol{r}_i, l_i)\}_{i=1}^{n}$ be the set of all objects that appears in image $\boldsymbol{I}$ of any category in $\mathcal{L}$. $\boldsymbol{r}_i \in \mathbb{R}^4$ and $l_i \in \mathcal{L}$ are the bounding box and the category label, respectively, of the i-th object. Then, given a pair $(\boldsymbol{I}, S)$ composed of an image and a set S of WordNet concepts, and a *concept depth* d, the defined task produces the set:

$$
F(\boldsymbol{I}, S, d) = \{(\boldsymbol{r}, l) \mid (\boldsymbol{r}, l) \in G(\boldsymbol{I}) \wedge S \cap f^d(l) \neq \varnothing\}.
$$

Please, notice that the standard object detector task can be defined in the proposed framework as $F(\boldsymbol{I}, f(\mathcal{L}), 0)$.

7.2.2 Definition of a Baseline

As a baseline for solving the "Find-That" task, a standard object detector coupled with an ad-hoc post-processing algorithm (i.e., Figure 7.4, *Post Processing Selection* component) that selects only the subset of objects compatible with the user's interest is considered.

Thus, the baseline can be formalized as in the following. Given an image $\boldsymbol{I}$, if $P_B(\boldsymbol{I}) = \{(\boldsymbol{r}_i, l_i)\}_{i=1}^{n_p}$ is the set of n_p objects predicted by an object detector, the baseline method estimates $F(\boldsymbol{I}, S, d)$ by $F_B(\boldsymbol{I}, S, d)$, as:

$$F_B(\boldsymbol{I}, S, d) = \{(\boldsymbol{r}, l) \mid (\boldsymbol{r}, l) \in P_B(\boldsymbol{I}) \wedge S \cap f^d(l) \neq \varnothing\}.$$

The post-processing algorithm checks if $S \cap f^d(l) \neq \varnothing$.

7.2.3 Improving on the Baseline: The new Model Proposal

The baseline can be improved by exploiting an object detector conditioned by the input WordNet concepts. Given an image jointly with a set of WordNet concepts in input, during training, the object detector learns to detect only the desired objects. Hence, implicitly the model tries to learn a mapping function from the set of WordNet concepts to the categories of the object detector. This helps in improving the quality of proposals that the *Post-processing Selection* component receives in input.

Formally, given an image $\boldsymbol{I}$ and a set S of concepts, if $P_C(\boldsymbol{I}, S) = \{\boldsymbol{r}_i, l_i\}$ is the set of objects predicted by a concept-conditioned object detector, $F(\boldsymbol{I}, S, d)$ is estimated by $F_C(\boldsymbol{I}, S, d)$, where:

$$F_C(\boldsymbol{I}, S, d) = \{(\boldsymbol{r}, l) \mid (\boldsymbol{r}, l) \in P_C(\boldsymbol{I}, S) \wedge S \cap f^d(l) \neq \varnothing\}.$$

In the following, more details on the architecture of the concept-conditioned object detector are given, as well as on the corresponding training procedure.

7.2.3.1 Model Architecture

Figure 7.6 presents an in-depth zoom on the *Concept-Conditioned Object Detector* block presented in Figure 7.4. It illustrates the proposed architecture that exploits the information given by the WordNet concepts during object detection. In fact, both an *Image* and a set of *WordNet Concepts* are provided in input to the model. The blocks that are components of a standard object detector, i.e., components that are defined by a meta-architecture (e.g., Faster

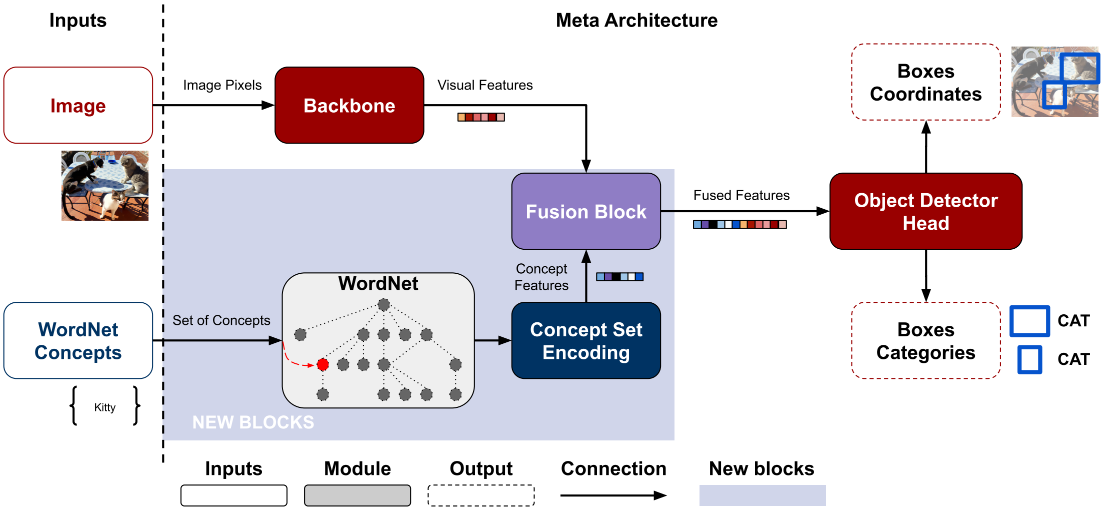


Figure 7.6: Overview of the concept-conditioned object detector. The *Image* with the *WordNet Concepts* are used as input to the model. The *Backbone* extracts the visual features from the image, while the *Concept Set Encoding* encodes in an embedding space the set of WordNet concepts in input. The concepts in input are highlighted in red in the *WordNet* block. Finally, the *Fusion Block* fuses the visual and concept features together and gives them as input to the *Object Detector Head*, which predicts the *Boxes Coordinates* and the *Boxes Categories* in output.

R-CNN or RetinaNet) and a backbone (e.g., ResNet-50 or ResNet-101), are depicted using the red color, while the background in light-blue color delimits the new blocks added to condition the object detector with concepts. The *Backbone* extracts the visual features from the image, while the *Concept Set Encoding* encodes in an embedding space the **set** of WordNet concepts in input. Finally, the *Fusion Block* fuses the visual and concept features together and sends them as input to the *Object Detector Head*, which predicts the *Boxes Coordinates* and the *Boxes Categories* in output.

7.2.3.2 Model Training

The model training can be performed by a standard end-to-end gradient-based procedure, however, the main issue is the lack of datasets compliant with the task definition, i.e., examples in the form $((\boldsymbol{I}, S), F(\boldsymbol{I}, S, d))$. For this reason, in this section, it is proposed an automatic procedure to derive an ad-hoc dataset D_F starting from an existing dataset D for object detection, which contains ground truth annotations for each object contained in each image $\boldsymbol{I}$, i.e., $G(\boldsymbol{I})$.

In order to define $(\boldsymbol{I}, S)$ and $F(\boldsymbol{I}, S, d)$, one needs to specify the "intent" S at concept depth d. This can be done recursively, starting from the base case $d = 0$, i.e., S_0, and then defining S_{d+1} starting from S_d. Let's start by defining S_0. Given an image $\boldsymbol{I}$ in D, it can be automatically generated the power set $\xi(\boldsymbol{I})$ of $G(\boldsymbol{I})$, i.e., the set of all the possible combinations of ground truths. Then, for each $E \in \xi(\boldsymbol{I})$, the set $L_E = \{l \mid (\boldsymbol{r}, l) \in E\}$ of all categories appearing in E is defined. At this point $\forall l \in L_E$:

$$S_0 = \{f^0(l)\}, \quad S_{d+1} = \{f^{d+1}(l)\}.$$

It could be disputed that the above procedure is not correct in the case in which a child of a concept does not find a match with a pre-defined object detection category. For example, consider the concept "siamese cat" and an object detector that only supports the category "CAT". In this case, since f("CAT") returns the concept "cat", i.e., a parent of "siamese cat", one may run the risk of generating an example involving an image that portrays a cat that is not a siamese cat jointly with the concept "siamese cat". However, such a query could actually be placed by a user that is not aware, as she/he shouldn't be, of the pre-defined object detection categories, and returning a bounding box containing a non-siamese cat is the best approximation that the object detector can do. It is a problem of the object detector: the more pre-defined categories the object detector can deal with, the better the performance of the system will be.

The power set approach $\xi(\boldsymbol{I})$ described above, however, generates an exponential number of training examples, and in practice, it is not a viable approach. For this reason, in this work, $\xi(\boldsymbol{I})$ and S_d are sampled to obtain a reasonable amount of training examples. Specifically, given an image $\boldsymbol{I}$ with its ground truth annotations $G(\boldsymbol{I})$, the procedure that synthesizes the concepts in input starts by sampling uniformly from $\xi(\boldsymbol{I})$ a subset $\hat{\xi}(\boldsymbol{I})$ of its members. The reduced set of concepts $\hat{S}_0$ is then obtained from $\hat{\xi}(\boldsymbol{I})$. Starting from $\hat{S}_0$, the reduced set of concepts $\hat{S}_{d+1}$ is obtained by sampling the set S_{d+1}. Specifically, a concept for each object to search in the image is sampled uniformly. For example, given Figure 7.3, input three concepts are provided as input: one concept for the object labeled as "BOWL" and two sampled concepts associated with the objects labeled as "CAT", i.e., one for each cat occurring in the image.

7.2.4 Experimental Assessment

The conditioned object detectors are evaluated on datasets generated starting from two widely adopted datasets (i.e., COCO and Visual Genome), considering two object detector meta-architectures (i.e., RetinaNet [65] and DynamicHead [58]) and several backbones, such as ResNet-50, ResNet-101, and Swin-Tiny.

See Appendix E.3 for the models' implementation details, Appendix E.2 for the model selection performed in this work, and Appendix E.7 to inspect some qualitative examples made by both standard and concept-conditioned object detectors.

7.2.4.1 Datasets and Evaluation Metrics

The COCO dataset [128] is an 80-class common object detection dataset. It is used the 2017 version of the dataset. Since the COCO test set ground truths are not publicly available online, in this work, the available validation set is used to generate the test set, and 5K images are randomly selected from the training set to generate the validation set. In this section, the COCO test set always refers to the original COCO validation set. Moreover, the COCO validation set refers to the 5K examples sampled from the original COCO training set, and the COCO training set refers to the examples left after sampling.

The Visual Genome [32] dataset consists of 108077 images with 1600 classes. Every split of data is available online with its ground truth annotations. Hence, on this dataset, it is adopted the splits available online for training, validating, and testing the models. See Appendix E.1 for more details on the datasets considered in this section.

The procedure presented in Section 7.2.3.2 allows the generation of new datasets to train and evaluate concept-conditioned models when deployed for searching all the objects contained in the images, as well as just a subset of objects as specified by the input concepts. More in detail, for each original dataset, **two** more datasets (with all their splits) are generated. The first dataset aims to evaluate the object detector when searching for all the objects in the images. In other words, for each image, $\boldsymbol{I}$, the set S is composed of at least one concept related to each ground truth in $G(\boldsymbol{I})$. The second dataset, dubbed "Focused", aims to evaluate the object detector when searching for only a subset of objects in the images. For each example $((\boldsymbol{I}, G(\boldsymbol{I}))$, the procedure presented in Section 7.2.3.2 generates the example $((\boldsymbol{I}, S), F(\boldsymbol{I}, S, d))$, which focuses on just a subset of all the objects $G(\boldsymbol{I})$.

Note that the examples to use in input to the model during training are generated at "runtime", while during evaluation, the results are computed on a pre-calculated set of examples (i.e., validation and test sets are fixed for each dataset).

The following metrics are adopted to evaluate models' performances: (i) *mean Average Precision (AP)*: this metric is the mean Average Precision per class defined by the COCO dataset[12], and (ii) *AP50*: this metric is the mean Average Precision per class, defined by the

[12] https://cocodataset.org/\protect#\relaxdetection-eval

COCO dataset, that evaluates the AP metric only considering the Intersection over Union (IoU) threshold of 0.5. See Appendix A.3 for more details about the IoU metric. These are standard object detection metrics that, in the end, allow for a fair comparison of the proposed model on the object detection task to demonstrate the effectiveness of the proposed approach over standard object detectors.

7.2.4.2 Object Detector Architectures and Backbones

The proposed approach is evaluated considering two object detectors, namely RetinaNet [65] and DynamiHead [58]. To assess the effectiveness of the proposed approach, each model is evaluated considering more backbones: ResNet-50 [72], ResNet-101, and Swin-Tiny [156]. Each model adopts the Feature Pyramid Network [157] to extract image features at different resolutions. See Appendix E.3 for an overview of the number of neurons constituting each model.

Each object detector model is modified to be conditioned with the concepts, i.e., "Concept RetinaNet" and "Concept DynamicHead" are the proposed models that can exploit the user's intent during object detection.

7.2.4.3 Comparing Standard and Concept-Conditioned Object Detectors before Filtering

In this section, it is investigated the benefit of leveraging the user's intent directly in the object detector architectures. It is done by considering the quality of the output of the detectors, before filtering, i.e., the *Post-processing Selection* component. Moreover, this section assesses concept-conditioned object detectors and standard object detectors in detecting all the objects depicted in the images, i.e., $G(\boldsymbol{I})$. For concept-conditioned models, the set of WordNet concepts used to condition the object detection process is built appropriately to include a concept for each object present in the image ground truth annotations.

Table 7.6 presents the results obtained by the object detectors when they are deployed for searching all the objects contained in COCO and in Visual Genome datasets. *AP (%)* refers to the object detection mean Average Precision, while *AP50 (%)* refers to the object detection mean Average Precision with *IoU* ≥ 0.5. The models conditioned with the concepts are highlighted with the dove gray color. In addition, the AP metric is also evaluated by considering several bounding box dimensions. The threshold values are defined by the COCO

Meta Architecture	Backbone	COCO (AP%)					Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
RetinaNet	ResNet-50	36.4	20.7	40.0	47.5	56.0	3.5	2.1	3.8	5.0	7.0
Concept RetinaNet	ResNet-50	39.4	26.2	43.1	51.1	61.3	4.4	2.6	4.7	5.8	9.0
RetinaNet	ResNet-101	38.7	22.1	42.9	50.5	58.4	3.9	2.3	4.0	5.6	7.6
Concept RetinaNet	ResNet-101	41.3	26.1	45.6	53.1	63.5	4.4	2.7	4.6	6.0	9.0
DynamicHead	ResNet-50	44.1	27.2	47.8	57.2	61.9	5.6	3.1	5.8	8.1	10.0
Concept DynamicHead	ResNet-50	50.2	33.5	53.9	65.1	71.5	9.3	5.3	9.3	12.6	17.5
DynamicHead	ResNet-101	44.1	26.7	47.7	57.2	61.6	5.7	2.9	6.1	8.2	10.1
Concept DynamicHead	ResNet-101	50.2	33.9	54.3	64.8	71.6	9.6	5.2	9.8	13.1	18.0
DynamicHead	Swin-Tiny	49.9	32.8	53.7	63.9	68.4	6.7	3.6	7.0	9.7	11.7
Concept DynamicHead	Swin-Tiny	54.7	37.8	58.5	68.1	76.0	10.4	6.1	11.0	13.8	19.3

Table 7.6: Object detection results on datasets generated from COCO and Visual Genome with $d = 1$, and considering several bounding boxes dimensions. *AP (%)* refers to the object detection *mean Average Precision*, while *AP50 (%)* refers to the object detection *mean Average Precision* with *IoU* ≥ 0.5. The models conditioned with the concepts are highlighted with the dove gray color.

dataset[13]: (i) *Small* refers to bounding boxes whose area is less than 32^2 pixels; (ii) *Medium* refers to bounding boxes whose area is less than 96^2 pixels and larger than 32^2; and (iii) *Large* refers to bounding boxes whose area is larger than 96^2 pixels. *All* refers to the case in which the evaluation is performed considering all the bounding boxes. In COCO, approximately 41% of the boxes are small size, approximately 34% of the boxes are medium size, and approximately 24% of boxes are large size.

Noticeably, the proposed concept-conditioned models, exploiting the user's intent, always perform better than standard object detectors when they are deployed for searching all the objects depicted in an image. Concept DynamicHead achieves the best outcomes in both datasets with ResNet as the backbone. Overall, the improvements given by Concept DynamicHead over the standard DynamicHead models are higher than the improvement of Concept RetinaNet over standard RetinaNet. On COCO, the larger AP improvement (6.1%) is given by Concept DynamicHead (50.2%) over DynamicHead (44.1%), both with ResNet-101/50. Even on Visual Genome, the same architecture and backbone give the best improvements (3.9% for ResNet-101).

Unexpectedly, the DynamicHead model coupled with the ResNet-50 and ResNet-101 backbones performs similarly. Given the higher neural network expressivity given by the Resnet-101 over ResNet-50, allowed by the largest number of parameters that amounts to 57.8M, the model outcomes should be better than those of ResNet-50. This is likely due to a non-exhaustive model selection performed on COCO and Visual Genome, which is detailed in Appendix E.2.

Regarding the AP metric evaluated according to the bounding boxes dimension (i.e., Small, Medium, and Large columns), it is visible that the concept-conditioned models benefit mostly in detecting small objects in COCO and large objects in Visual Genome.

In conclusion, it is evident that whenever the user's intent is exploited to condition the object detector architectures, their detection performance always increases. More results adopting the *Post-processing Selection* component in searching for all the objects in the image are detailed in Appendix E.5.

During model training, the maximum *concept depth* d value considered during the WordNet sampling process plays a fundamental role in the proposed approach. High-depth values force the model to learn more relations among categories and WordNet concepts, making the task that the model has to solve more difficult. Conversely, a low-depth value makes the

[13] Bounding box dimensions defined by the COCO dataset: `https://cocodataset.org/#detection-eval`

Depth Value	AP (%)	AP50 (%)	N. of Concepts
0	39.8	62.1	80
1	39.4	61.3	954
2	39.5	61.5	2586
3	39.2	61.1	5054
4	39.4	61.1	7274

Table 7.7: Object detection results on COCO test set varying the concept depth values used for generating the WordNet concepts. The values are obtained using the proposed concept-conditioned RetinaNet model with ResNet-50.

learning task easier while constraining the generalization of the proposed approach to only a small set of concepts.

Table 7.7 highlights the impact of employing different depth values on the conditioned models. The results were obtained with RetinaNet, using ResNet-50 as the backbone, on the COCO test set. AP and AP50 are the metrics adopted for the evaluation of the models. In this experiment, the concepts are not sampled, so depth value 0 refers to examples involving concepts in S_0, depth value 1 refers to examples involving concepts in S_1, and so on. As can be seen from the table, the best AP result is obtained with a depth value of 0, and there is no abrupt deterioration in the results, increasing the depth value from 0 to 4. More in detail, from the depth value of 0 to 1, the deterioration in the AP metric amounts to 0.4%, the same value with respect to depth 4. The largest drop in performance is observed for depth 3, where it reaches 0.6%. Notice that the number of concepts the user can adopt to express her/his intent grows significantly from depth 0 to depth 4.

In conclusion, these results suggest that in the COCO dataset, it is possible to generalize the model to the use of 7274 different WordNet concepts trading off some of the effectiveness of the model.

7.2.4.4 Searching for a Subset of Objects

This section compares concept-conditioned object detectors and standard object detectors, *coupled* with the *Post-processing Selection* component, to search for just a *subset* of objects depicted in the images and consistent with the input concepts. To this aim, all the models are assessed on the datasets generated as explained in Section 7.2.3.2 that is dub "Focused COCO" and "Focused Visual Genome".

Table 7.8 presents the obtained results. From this table, it can be seen that the proposed conditioned models outperform standard object detectors in all architectures and backbones

Meta Architecture	Backbone	Focused COCO (AP%)					Focused Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
RetinaNet	ResNet-50	40.6	25.9	43.6	52.1	63.3	6.9	4.2	6.7	9.1	13.5
Concept RetinaNet	ResNet-50	42.1	29.0	45.1	53.8	65.9	7.3	4.1	7.3	9.6	14.8
RetinaNet	ResNet-101	43.1	27.3	46.9	55.1	65.8	7.3	4.3	7.4	9.7	14.3
Concept RetinaNet	ResNet-101	43.9	29.3	47.9	55.5	67.5	7.4	4.5	7.3	9.7	14.8
DynamicHead	ResNet-50	49.0	31.7	51.9	62.7	69.2	10.7	5.6	10.4	14.1	18.8
Concept DynamicHead	ResNet-50	52.1	35.9	55.0	66.5	73.7	13.2	7.4	13.0	17.6	24.3
DynamicHead	ResNet-101	49.2	32.0	52.6	62.6	69.3	10.9	5.4	10.5	14.6	19.3
Concept DynamicHead	ResNet-101	52.2	35.9	55.3	66.6	73.7	13.7	7.7	13.3	18.3	25.1
DynamicHead	Swin-Tiny	54.3	37.7	57.2	67.4	74.5	12.7	6.6	12.0	17.0	21.9
Concept DynamicHead	Swin-Tiny	56.5	40.5	59.3	69.7	77.9	14.7	8.5	14.6	19.3	26.9

Table 7.8: Object detection results on Focused COCO and Focused Visual Genome datasets with $d = 1$, considering several bounding box dimensions. *AP (%)* refers to the object detection *mean Average Precision*, while *AP50 (%)* refers to the object detection *mean Average Precision* with *IoU* ≥ 0.5. The models conditioned with the concepts are highlighted with the dove gray color.

combinations.

Both datasets achieve the best AP results by deploying DynamicHead with ResNet backbones. On Focused COCO, the larger AP improvement (3.1%) is given by Concept DynamicHead (52.1%) over DynamicHead (49.0%), both with ResNet-50. While, on Visual Genome, the larger AP improvement (2.8%) is given by Concept DynamicHead (13.7%) over DynamicHead (10.9%), both with ResNet-101. In general, the improvements achieved on the Focused COCO dataset by the conditioned models are higher than those achieved in the Focused Visual Genome.

In conclusion, conditioning the object detection with the user's intent generally improves the detection performance of an object detector that adopts a post-processing procedure for selecting the boxes of interest.

See Appendix E.6 to explore results by class.

7.2.5 Related Works

This work is mainly related to the object detection area of research, as object detectors can be adapted for solving the proposed "Find-That" task. According to the works in literature, only Fornoni *et al.* [158] proposed a work that is similar to this work. The authors aim to condition object detectors with prior information (as done in this section), emphasizing mainly object detectors with efficient constraints (mobile). They re-use existing object detector code with minor changes, and they also developed a procedure to generate the user's prior intent from the ground truths available in existing object detection datasets. However, there is a significant difference in how the user's intent is represented. In [158], the object detector is modified to consider an input composed of images and categories augmented with spatial information needed to constrain the object search in the image. The categories are those defined by the dataset, and their model is conditioned with a vector of ones (to search) and zeroes (not to search) for each class (i.e., an 80-dimension vector for COCO). In the approach proposed in this section, the model is conditioned in input with WordNet concepts instead of categories, and the concepts are not augmented with the spatial location information, even if the model can be easily extended to do so. Hence, their approach **does not** tackle the mismatch problem between the concepts expressed by the user and the classes of the object detector, thus solving an easier problem compared to the proposed task addressed in this section. In addition, since the target label is provided as an input to their proposed conditional model, their approach only aims to localize the objects in the image. For this reason, their evaluation is category-agnostic. Instead, in the work proposed in this section, the concept-conditioned models aim to locate and correctly classify the objects depicted in the image. Thus, the evaluation performed is category-aware. A direct experimental comparison versus the above approach is not possible since: (i) their approach uses different prior information than that used in this section (i.e., category vectors of ones and zeros versus WordNet embeddings); (ii) their evaluation setting (online style) significantly differs from that adopted in this work (fixed test set); (iii) their code is not available online, making impossible an evaluation in a setting comparable with the one used in this section.

This work is also related to other research areas, as often object detection is used as a building block for solving many other downstream tasks, such as Referring Expression [6, 7, 8, 9, 10, 11], also known as Visual-Textual Grounding, Visual Question Answering [12, 13, 14], Visual-Textual-Knowledge Entity Linking (VTKEL) [15, 16, 17] and Image-Text Retrieval [18, 159, 160, 19]. Note that the proposed approach could be deployed to solve the VTKEL problem, conditioning the object detector with the entities of the knowledge graph. See Chapter 3 for both the Visual-Textual Grounding and VTKEL State-of-the-Art.

The defined "Find-That' task resembles the referring expression task, although there are substantial differences. First of all, the user's intent needs to be represented as a textual phrase, while in the model proposed in this section, the user's intent is expressed with one or more WordNet [33] concepts. Secondly, following the current State-of-the-Art, referring expression models predict only the bounding box that best matches the textual phrase in the output. For this reason, when the user's intent concerns multiple different objects depicted in the image, multiple independent queries should be performed to retrieve all objects of interest. In addition, when the user's intent concerns multiple objects of the same type, the referring expression approach is no longer suitable. Lastly, for training, referring expression models need to use detailed datasets comprising images, boxes coordinate, textual phrases, and the ground truth of the corresponding bounding box in the image for each phrase. These annotations are difficult to collect, so the referring expression datasets contain fewer examples than those of detection. Appendix E.8 reports a more detailed comparison.

7.2.6 Conclusion and Future Works

This section proposed a novel approach to focused object search in an image by conditioning existing object detectors with the user's search intent, represented as a set of WordNet concepts. The proposed approach can be implemented with minor changes to a standard object detector, it does not require the modification or addition of any object detector loss and contributes to the estimation of the probability distribution $\mathbb{P}\left(X_{\boldsymbol{b},V} \mid X_{\boldsymbol{q},V}, \boldsymbol{Z_b}, \widehat{\boldsymbol{Z_b}}, KG\right)$ presented in Section 4.4.

The proposed concept-conditioned object detector can be trained on existing datasets for object detection without the need to add or modify existing annotations to consider the WordNet concepts. The approach is tested for searching all objects on COCO and Visual Genome datasets and also for searching just subsets of objects using the newly defined Focused COCO and Focused Visual Genome datasets. Several object detector architectures and backbones are considered, showing that the proposed concept-conditioned object detector approach performs better than the standard object detector baseline.

Since concept-conditioned models adopt pre-computed WordNet embedding representations, future work aims to evaluate the model performance using different embeddings weights and to improve the fusion of the multimodal information in the object detector architecture. Future work will also extend the WordNet concepts with entities that belong to heterogeneous knowledge graphs, such as YAGO [96, 97, 98]. Finally, future work aims to integrate this model within a word sense disambiguation system, with the goal of solving multimodal text-image tasks, such as visual question answering and visual-textual grounding.

8
Conclusions and Future Works

In conclusion, this Ph.D. thesis aims to improve visual-textual grounding tasks by introducing a novel approach that incorporates a third modality, in the form of a graph, alongside the traditional image and text modalities. This graph-based approach is expected to enhance the performance and accuracy of the visual-textual grounding models.

For this reason, Chapter 4 presented a formal probabilistic framework developed to analyze the integration of the three modalities and to deal with the inherent uncertainties in solving visual-textual grounding tasks. The framework allows the analysis of the already published works, highlighting their strengths and weaknesses according to how the modalities are adopted in the model. Moreover, it constitutes an important tool that can be employed to devise a novel approach to visual-textual grounding based on an innovative factorization of probabilities not yet explored in the literature.

Furthermore, this thesis also investigates improvements to the traditional two-modality visual-textual grounding task. Two contributions are proposed. The first, introduced in Chapter 5, presents a new loss function for training two-stage models in a supervised setting. The novel loss combines a grounding loss and a bounding box coordinates refinement loss, both based on the probability distribution over the set of pre-defined classes returned by the object detector. Experiments have proven that when a model adopts the new loss, it reaches better results. Nevertheless, the work proposed in Chapter 5 has also some limitations, such as the fact that it adopts a simple multi-modal feature fusion component and that the new loss function relies on the cosine similarity to calculate the similarity between the predicted class probabilities of the bounding boxes (i.e., matrix $\boldsymbol{C}$ defined in Section 5.3.0.2). Future works aim to adopt more sophisticated multi-modal feature fusion components, such as bilinear-pooling [78], and to explore different similarity functions such as the Jeffreys divergence [161]

The second contribution, introduced in Chapter 6, presents a two-stage model tackling the weakly-supervised visual-textual grounding. The proposed model is based on the principle of first predicting a rough alignment among phrases and boxes adopting a module that does not require training, and then refining those alignments using a learnable neural network. The model is trained to maximize the multimodal similarity between an image and a sentence describing that image while minimizing the multimodal similarity of the same sentence and a new unrelated image, carefully selected to help as much as possible during training. In light of this, the approach presented a State-of-the-Art performance on two well-established visual-textual grounding datasets. However, the module that does not require training is sensible to the pre-trained GloVe [3] embeddings which are used to initialize its weights. Indeed, the less the pre-training embeddings are able to gather the semantic information between pairs of words, the less accurate this module's predictions will be. Future works in this direction will adopt more recent embeddings such as those of BERT [43].

Chapter 7 introduced two potential extensions of the presented approaches. More in particular, in Section 7.1, this thesis addresses the issue of noisy class labels in the commonly used Bottom-Up Faster R-CNN object detector, proposing a set of less noisy labels. Indeed, the object detector classes are important for solving the visual-textual grounding problem, especially when considering graph information. The results have shown that the object detector trained on the new cleaned labels performs better than the object detector trained on the noisy set of labels. Moreover, the utility of the new model is evaluated as a black-box feature extractor for a downstream visual-textual grounding task with the Bilinear Attention Network model (BAN) [4]. The results show that features from the new object detector can consistently improve the BAN model across commonly used object detection thresholds. However, the new set of labels is made by a skilled human annotator, which may have introduced some unwanted human bias or error into the cleaning process. Future works could generate alternative cleaned categories by merging similar ones, e.g., using a framework similar to Confidence Learning [153].

Last but not least, Section 7.2 introduced a novel approach to focused object search in an image by conditioning existing object detectors with the user's search intent, represented as a set of WordNet concepts. The proposed approach can be implemented with minor changes to a standard object detector, it does not require the modification or addition of any object detector loss and contributes to the estimation of the novel probability factorization presented in Section 4.4. According to the evaluation performed, the proposed concept-conditioned object detectors achieve better results than standard object detectors. However, the proposed concept-conditioned models consider only the subset of the WordNet graph defined by nodes related by the parent-child relations. Future works aim to use additional relations as well.

Overall, the research presented in this Ph.D. thesis contributes to the understanding and advancement of visual-textual grounding techniques. The tools and insights offered in this document hold the potential to facilitate the development of more accurate and efficient visual-textual grounding models.

Future works aim to fully estimate the probability factorization presented in Section 4.4, solving the visual-textual grounding by adopting the loss function introduced in Chapter 5, the concept-conditioned object detector presented in Section 7.2, and the clean set of classes proposed in Section 7.1. In addition, future works aim to incorporate the conditioned object detector presented in Section 7.2 into the model presented in Chapter 6, aspiring to generate better bounding boxes proposal than those generated with the Bottom-Up Faster R-CNN. Moreover, future works aim to integrate the object detector trained on the new cleaned classes in the models presented in Chapter 5 and Chapter 6. To conclude, other future works aim to use the framework presented in Chapter 4 to design and estimate a new novel probability distribution factorization to adopt in solving the visual-textual grounding task.

References

[1] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993. [Online]. Available: https://aclanthology.org/J93-2004

[2] J. Nivre, Ž. Agić, L. Ahrenberg, L. Antonsen, M. J. Aranzabe, M. Asahara, L. Ateyah, M. Attia, A. Atutxa, L. Augustinus *et al.*, "Universal dependencies 2.1," 2017.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[4] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," *Advances in neural information processing systems*, vol. 31, 2018.

[5] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[6] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 817–834.

[7] K. Chen, J. Gao, and R. Nevatia, "Knowledge aided consistency for weakly supervised phrase grounding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4042–4050.

[8] H. Zhang, Y. Niu, and S.-F. Chang, "Grounding referring expressions in images by variational context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4158–4166.

[9] T. Gupta, A. Vahdat, G. Chechik, X. Yang, J. Kautz, and D. Hoiem, "Contrastive learning for weakly supervised phrase grounding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 752–768.

[10] D. Rigoni, L. Serafini, and A. Sperduti, "A better loss for visual-textual grounding," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 49–57.

[11] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "MDETR - modulated detection for end-to-end multi-modal understanding," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 1760–1770. [Online]. Available: https://doi.org/10.1109/ICCV48922.2021.00180

[12] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015, pp. 2425–2433.

[13] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016, pp. 4613–4621.

[14] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[15] S. Dost, L. Serafini, M. Rospocher, L. Ballan, and A. Sperduti, "Vtkel: a resource for visual-textual-knowledge entity linking," in *ACM*, 2020, pp. 2021–2028.

[16] ——, "Jointly linking visual and textual entity mentions with background knowledge," in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2020, pp. 264–276.

[17] ——, "On visual-textual-knowledge entity linking," in *ICSC*. IEEE, 2020, pp. 190–193.

[18] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[19] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[20] Q. Zhang, Z. Lei, Z. Zhang, and S. Z. Li, "Context-aware attention network for image-text retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[21] W.-H. Li, S. Yang, Y. Wang, D. Song, and X.-Y. Li, "Multi-level similarity learning for image-text retrieval," *Information Processing & Management*, vol. 58, no. 1, p. 102432, 2021.

[22] S. Wang, R. Wang, Z. Yao, S. Shan, and X. Chen, "Cross-modal scene graph matching for relationship-aware image-text retrieval," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[23] Z. Yu, J. Yu, C. Xiang, Z. Zhao, Q. Tian, and D. Tao, "Rethinking diversified and discriminative proposal generation for visual grounding," in *IJCAI*, J. Lang, Ed. ijcai.org, 2018, pp. 1114–1120.

[24] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, "A fast and accurate one-stage approach to visual grounding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4683–4693.

[25] A. Sadhu, K. Chen, and R. Nevatia, "Zero-shot grounding of objects from natural language queries," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4694–4703.

[26] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*. Springer, 2014, pp. 391–405.

[27] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[28] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *NeurIPS*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 91–99.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[31] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[32] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[33] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[34] M. Sanderson, "Christopher d. manning, prabhakar raghavan, hinrich schütze, introduction to information retrieval, cambridge university press. 2008. isbn-13 978-0-521-86571-5, xxi+ 482 pages." *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[35] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[36] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton, "Quantitative evaluation of passage retrieval algorithms for question answering," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 41–47.

[37] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 384–394.

[38] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[40] 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.

[41] S. T. Dumais *et al.*, "Latent semantic analysis," *Annu. Rev. Inf. Sci. Technol.*, vol. 38, no. 1, pp. 188–230, 2004.

[42] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[43] 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.

[44] L. Marquez, L. Padro, and H. Rodriguez, "A machine learning approach to pos tagging," *Machine Learning*, vol. 39, no. 1, pp. 59–91, 2000.

[45] M.-C. De Marneffe, B. MacCartney, C. D. Manning *et al.*, "Generating typed dependency parses from phrase structure parses." in *Lrec*, vol. 6, 2006, pp. 449–454.

[46] M.-C. De Marneffe and C. D. Manning, "The stanford typed dependencies representation," in *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, 2008, pp. 1–8.

[47] N. Silveira, T. Dozat, M.-C. De Marneffe, S. R. Bowman, M. Connor, J. Bauer, and C. D. Manning, "A gold standard dependency corpus for english." in *LREC*. Citeseer, 2014, pp. 2897–2904.

[48] Y. Lin, J.-B. Michel, E. A. Lieberman, J. Orwant, W. Brockman, and S. Petrov, "Syntactic annotations for the google books ngram corpus," in *Proceedings of the ACL 2012 system demonstrations*, 2012, pp. 169–174.

[49] D. Zeman, "Reusable tagset conversion using tagset drivers." in *LREC*, vol. 2008, 2008, pp. 28–30.

[50] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira *et al.*, "Universal dependencies v1: A multilingual treebank collection," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 1659–1666.

[51] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[52] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "Flair: An easy-to-use framework for state-of-the-art nlp," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 54–59.

[53] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. (2020) spacy: Industrial-strength natural language processing in python. [Online]. Available: https://spacy.io/

[54] M. Straka, "Udpipe 2.0 prototype at conll 2018 ud shared task," in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2018, pp. 197–207.

[55] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A python natural language processing toolkit for many human languages," *arXiv preprint arXiv:2003.07082*, 2020.

[56] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[57] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9759–9768.

[58] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7373–7382.

[59] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[60] X. Wang, S. Zhang, Z. Yu, L. Feng, and W. Zhang, "Scale-equalizing pyramid convolution for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 359–13 368.

[61] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9657–9666.

[62] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-end semi-supervised object detection with soft teacher," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3060–3069.

[63] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[64] J. Yang, C. Li, and J. Gao, "Focal modulation networks," *arXiv preprint arXiv:2203.11926*, 2022.

[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[66] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[67] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[68] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2015.

[69] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[70] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[71] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, p. 1137, 2017.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[74] K. Chen, R. Kovvuri, and R. Nevatia, "Query-guided regression network with context policy for phrase grounding," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 824–832.

[75] M. Engilberge, L. Chevallier, P. Pérez, and M. Cord, "Deep semantic-visual embedding with localization," in *RFIAP 2018-Congrès Reconnaissance des Formes, Image, Apprentissage et Perception*, 2018.

[76] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *ICCV*. IEEE Computer Society, 2017, pp. 1946–1955.

[77] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.

[78] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *EMNLP*, J. Su, X. Carreras, and K. Duh, Eds. The Association for Computational Linguistics, 2016, pp. 457–468.

[79] D.-K. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *CVPR*, 2018, pp. 6087–6096.

[80] H. Akbari, S. Karaman, S. Bhargava, B. Chen, C. Vondrick, and S.-F. Chang, "Multilevel multimodal common semantic space for image-phrase grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 476–12 486.

[81] M. Bajaj, L. Wang, and L. Sigal, "G3raphground: Graph-based language grounding," in *ICCV*, 2019, pp. 4281–4290.

[82] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4555–4564.

[83] S. Datta, K. Sikka, A. Roy, K. Ahuja, D. Parikh, and A. Divakaran, "Align2ground: Weakly supervised phrase grounding guided by image-caption alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2601–2610.

[84] S. A. Javed, S. Saxena, and V. Gandhi, "Learning unsupervised visual grounding through semantic self-supervision," *arXiv preprint arXiv:1803.06506*, 2018.

[85] F. Zhao, J. Li, J. Zhao, and J. Feng, "Weakly supervised phrase localization with multiscale anchored transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5696–5705.

[86] X. Liu, L. Li, S. Wang, Z.-J. Zha, D. Meng, and Q. Huang, "Adaptive reconstruction network for weakly supervised referring expression grounding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2611–2620.

[87] A. Arbelle, S. Doveh, A. Alfassy, J. Shtok, G. Lev, E. Schwartz, H. Kuehne, H. B. Levi, P. Sattigeri, R. Panda *et al.*, "Detector-free weakly supervised grounding by separation," *arXiv preprint arXiv:2104.09829*, 2021.

[88] Q. Wang, H. Tan, S. Shen, M. W. Mahoney, and Z. Yao, "Maf: Multimodal alignment framework for weakly-supervised phrase grounding," *arXiv preprint arXiv:2010.05379*, 2020.

[89] J. Wang and L. Specia, "Phrase localization without paired training examples," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4663–4672.

[90] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[91] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7206–7215.

[92] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[93] S. Fidler, A. Sharma, and R. Urtasun, "A sentence is worth a thousand pixels," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1995–2002.

[94] F. Xiao, L. Sigal, and Y. Jae Lee, "Weakly-supervised visual grounding of phrases with linguistic structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5945–5954.

[95] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? text-to-image coreference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3558–3565.

[96] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "Yago2: A spatially and temporally enhanced knowledge base from wikipedia," *Artificial intelligence*, vol. 194, pp. 28–61, 2013.

[97] F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *7th biennial conference on innovative data systems research*. CIDR Conference, 2014.

[98] T. Pellissier Tanon, G. Weikum, and F. Suchanek, "Yago 4: A reason-able knowledge base," in *European Semantic Web Conference*. Springer, 2020, pp. 583–596.

[99] F. Corcoglioniti, M. Rospocher, and A. P. Aprosio, "Frame-based ontology population with pikes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3261–3275, 2016.

[100] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[101] M. Bevilacqua, T. Pasini, A. Raganato, and R. Navigli, "Recent trends in word sense disambiguation: A survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc, 2021.

[102] S. Kumar, S. Jat, K. Saxena, and P. Talukdar, "Zero-shot word sense disambiguation using sense definition embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5670–5681.

[103] M. Bevilacqua and R. Navigli, "Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2854–2864.

[104] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *arXiv preprint arXiv:2005.03572*, 2020.

[105] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[106] Z. Chen, P. Wang, L. Ma, K.-Y. K. Wong, and Q. Wu, "Cops-ref: A new dataset and task on compositional referring expression comprehension," in *CVPR*, June 2020.

[107] 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.

[108] 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.

[109] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.

[110] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "Referit game: Referring to objects in photographs of natural scenes," in *EMNLP*, 2014.

[111] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng, "Structured matching for phrase localization," in *ECCV*. Springer, 2016, pp. 696–711.

[112] L. Wang, Y. Li, J. Huang, and S. Lazebnik, "Learning two-branch neural networks for image-text matching tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 394–407, 2018.

[113] R. A. Yeh, J. Xiong, W. W. Hwu, M. N. Do, and A. G. Schwing, "Interpretable and globally optimal prediction for textual grounding using image concepts," in *NeurIPS*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1912–1922.

[114] P. Dogan, L. Sigal, and M. Gross, "Neural sequential phrase grounding (seqground)," in *CVPR*, 2019, pp. 4175–4184.

[115] B. A. Plummer, P. Kordas, M. H. Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik, "Conditional image-text embedding networks," in *ECCV*, 2018, pp. 249–264.

[116] Y. Liu, B. Wan, X. Zhu, and X. He, "Learning cross-modal context graph for visual grounding," in *AAAI*. AAAI Press, 2020, pp. 11 645–11 652.

[117] J. Liu and J. Hockenmaier, "Phrase grounding by soft-label chain conditional random field," in *EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 5111–5121.

[118] F. Wu, Z. Xu, and Y. Yang, "An end-to-end approach to natural language object retrieval via context-aware deep reinforcement learning," *arXiv preprint arXiv:1703.07579*, 2017.

[119] J. Li, Y. Wei, X. Liang, F. Zhao, J. Li, T. Xu, and J. Feng, "Deep attribute-preserving metric learning for natural language object retrieval," in *Proceedings of the 2017 ACM*

on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017, Q. Liu, R. Lienhart, H. Wang, S. K. Chen, S. Boll, Y. P. Chen, G. Friedland, J. Li, and S. Yan, Eds. ACM, 2017, pp. 181–189.

[120] H. Jiang, Y. Lin, D. Han, S. Song, and G. Huang, "Pseudo-q: Generating pseudo language queries for visual grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 513–15 523.

[121] Y. Liu, B. Wan, L. Ma, and X. He, "Relation-aware instance refinement for weakly supervised visual grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5612–5621.

[122] Z. Fang, S. Kong, C. Fowlkes, and Y. Yang, "Modularized textual grounding for counterfactual resilience," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6378–6388.

[123] L. Wang, J. Huang, Y. Li, K. Xu, Z. Yang, and D. Yu, "Improving weakly supervised visual grounding by contrastive knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 090–14 100.

[124] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[125] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[126] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, J. Gao *et al.*, "Vision-language pre-training: Basics, recent advances, and future trends," *Foundations and Trends® in Computer Graphics and Vision*, vol. 14, no. 3–4, pp. 163–352, 2022.

[127] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[128] 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.

[129] D. Rigoni, D. Elliott, and S. Frank, "Cleaner categories improve object detection and visual-textual grounding," in *Image Analysis: 23rd Scandinavian Conference, SCIA 2023, Sirkka, Finland, April 18–21, 2023, Proceedings, Part I*. Springer, 2023, pp. 412–442.

[130] H. Zhang, P. Zhang, X. Hu, Y.-C. Chen, L. H. Li, X. Dai, L. Wang, L. Yuan, J.-N. Hwang, and J. Gao, "Glipv2: Unifying localization and vision-language understanding," *arXiv preprint arXiv:2206.05836*, 2022.

[131] Z.-Y. Dou, A. Kamath, Z. Gan, P. Zhang, J. Wang, L. Li, Z. Liu, C. Liu, Y. LeCun, N. Peng *et al.*, "Coarse-to-fine vision-language pre-training with fusion in the backbone," *arXiv preprint arXiv:2206.07643*, 2022.

[132] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, "Grounded language-image pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 965–10 975.

[133] Y. Yao, Q. Chen, A. Zhang, W. Ji, Z. Liu, T.-S. Chua, and M. Sun, "Pevl: Position-enhanced pre-training and prompt tuning for vision-language models," *arXiv preprint arXiv:2205.11169*, 2022.

[134] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[135] R. Cadene, C. Dancette, M. Cord, D. Parikh *et al.*, "Rubi: Reducing unimodal biases for visual question answering," *Advances in neural information processing systems*, vol. 32, 2019.

[136] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6281–6290.

[137] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 10 313–10 322.

[138] R. Cadene, H. Ben-Younes, M. Cord, and N. Thome, "Murel: Multimodal relational reasoning for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1989–1998.

[139] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8317–8326.

[140] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 13 041–13 049.

[141] C. Jing, Y. Jia, Y. Wu, X. Liu, and Q. Wu, "Maintaining reasoning consistency in compositional visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5099–5108.

[142] R. Wang, Y. Qian, F. Feng, X. Wang, and H. Jiang, "Co-vqa: Answering by interactive sub question sequence," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 2396–2408.

[143] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[144] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," *arXiv preprint arXiv:1908.08530*, 2019.

[145] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European Conference on Computer Vision*. Springer, 2020, pp. 121–137.

[146] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5100–5111.

[147] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 13–23. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html

[148] S. Frank, E. Bugliarello, and D. Elliott, "Vision-and-language or vision-for-language," *On Cross-Modal Influence in Multimodal Transformers.(2021). DOI: https://doi.org/10.18653/v1/2021. emnlp-main*, vol. 775, 2021.

[149] E. Bugliarello, R. Cotterell, N. Okazaki, and D. Elliott, "Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language berts," *arXiv preprint arXiv:2011.15124*, 2020.

[150] A. Mogadala, M. Kalimuthu, and D. Klakow, "Trends in integration of vision and language research: A survey of tasks, datasets, and methods," *Journal of Artificial Intelligence Research*, vol. 71, pp. 1183–1317, 2021.

[151] K. Kafle, R. Shrestha, and C. Kanan, "Challenges and prospects in vision and language research," *Frontiers in Artificial Intelligence*, vol. 2, p. 28, 2019.

[152] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19, 2022.

[153] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021.

[154] L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. v. d. Oord, "Are we done with imagenet?" [Online]. Available: https://arxiv.org/abs/2006.07159

[155] H. Wang, H. Wang, and K. Xu, "Categorizing concepts with basic level for vision-to-language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[156] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[157] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 936–944.

[158] M. Fornoni, C. Yan, L. Luo, K. Wilber, A. Stark, Y. Cui, B. Gong, and A. Howard, "Bridging the gap between object detection and user intent via query-modulation," *arXiv preprint arXiv:2106.10258*, 2021.

[159] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation," *arXiv preprint arXiv:1411.7399*, 2014.

[160] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *NeurIPS*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 2121–2129.

[161] I. Good, "Theory of probability harold jeffreys (447+ ix pp., oxford univ. press, 84 s.)," 1962.

[162] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[163] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206848

[164] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman,

Eds. AAAI Press, 2016, pp. 1955–1961. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484

[165] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.

Appendix A
Visual-Textual Grounding Datasets and Metrics

This appendix presents the main datasets and evaluation metrics adopted in the visual-textual grounding area of research. Follow the description of both the Flickr30k Entities [77] and ReferIt [110] datasets, and the definition of the Intersection over Union and Complete Intersection over Union [104] metrics.

A.1 Flickr30k Entities

The Flickr30k Entities dataset [77] is a widely used benchmark dataset in computer vision and natural language processing research for the task of visual-textual grounding, which involves associating words or phrases in natural language with corresponding objects or regions in visual data. The dataset consists of 31.783 images, each of which is accompanied by five captions that describe the image in natural language. Each caption includes a variable number of noun phrases that are associated with a set of bounding boxes ground truth coordinates.

Specifically, each caption in the dataset is annotated with a set of entity mentions, where each mention is associated with a bounding box indicating the region in the image that the entity refers to. The dataset contains 275K bounding boxes, 159K sentences, and 360K noun phrases. The standard split for training, validation, and test set as defined in [77], consisting of 30K, 1K, and 1K images, respectively.

The dataset has been used in numerous research studies and has contributed to significant advances in the field of visual-textual grounding and multimodal understanding.

A.2 ReferIt

The ReferIt [110] Game Entities dataset consists of 20.000 images from the MS COCO [128] dataset, each of which is accompanied by a set of referring expressions that describe specific objects or regions in the image. More in detail, the dataset contains 99K bounding boxes and 130K noun phrases, which were collected using an online game where players were asked to refer to objects in the images using natural language.

Each referring expression in the dataset is annotated with a bounding box indicating the region in the image that the expression refers to. The dataset also includes information about the annotator who provided each expression, including their nationality, native language, and proficiency in English. The standard split for training, validation, and test set as defined in [77], consisting of 9K, 1K, and 10K images, respectively.

The ReferIt Game Entities dataset is commonly used for training and evaluating models for the visual-textual grounding task, and has been used in numerous research studies. It has also led to significant advances in the development of models that can understand and interpret natural language expressions in the context of visual data.

This dataset differs from Flickr30k Entities since it does not contain sentences, meaning the noun phrases are mutually independent. For this reason, the State-of-the-Art models that depend on a sentence linking all the noun phrases, since they use a feature fusion operator that assumes the presence of the input sentence containing all the noun phrases, cannot be applied to it.

A.3 Intersection over Union (IoU)

Given a pair of bounding box coordinates $(\boldsymbol{b}_i, \boldsymbol{b}_j)$ with $\boldsymbol{b}_i, \boldsymbol{b}_j \in \mathbb{R}^4$, the *Intersection over Union*, also known as Jaccard index, is an evaluation metric used mainly in object detection tasks, which aims to evaluate how much the two bounding boxes refer to the same content in the image. It is defined as:

$$IoU(\boldsymbol{b}_i, \boldsymbol{b}_j) = \frac{|\boldsymbol{b}_i \cap \boldsymbol{b}_j|}{|\boldsymbol{b}_i \cup \boldsymbol{b}_j|}, \tag{A.1}$$

where $|\boldsymbol{b}_i \cap \boldsymbol{b}_j|$ is the area of the box obtained by the intersection of boxes $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, while $|\boldsymbol{b}_i \cup \boldsymbol{b}_j|$ is the area of the box obtained by the union of boxes $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$. It is invariant to the bounding boxes sizes, and it returns values that are strictly contained in the interval

$[0, 1] \subset \mathbb{R}$, where 1 means that the two bounding boxes refer to the same image area, while a score of 0 means that the two bounding boxes do not overlap at all. The fact that two bounding boxes that do not overlap have *IoU* score equal to 0, is the major issue of this metric: the zero value does not represent how much the two bounding boxes are far from each other. For this reason, in its standard definition, the *IoU* function is mainly used as an evaluation metric rather than as a component of a loss function for learning.

A.4 Complete Intersection over Union

In order to solve the issue of *IoU* when considering it as a loss function, [104] proposed the *Complete IoU* loss that is defined as:

$$\mathcal{L}_{CIoU}(\boldsymbol{b}_i, \boldsymbol{b}_j) = S\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) + D\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) + V\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) \tag{A.2}$$

$$S\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) = 1 - IoU(\boldsymbol{b}_i, \boldsymbol{b}_j); \tag{A.3}$$

$$D\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) = \frac{\rho\left(\boldsymbol{p_i}, \boldsymbol{p}_j\right)^2}{c^2}; \tag{A.4}$$

$$V\left(\boldsymbol{b}_i, \boldsymbol{b}_j\right) = \alpha \frac{4}{\pi^2}\left(\arctan\frac{wt_j}{ht_j} - \arctan\frac{wt_i}{ht_i}\right) \tag{A.5}$$

where $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ with $\boldsymbol{b}_i, \boldsymbol{b}_j \in \mathbb{R}^4$ are two bounding boxes, $\boldsymbol{p_i}$ and $\boldsymbol{p}_j$ are their central points, $IoU(\boldsymbol{b}_i, \boldsymbol{b}_j)$ is the standard *IoU*, ρ is the euclidean distance between the given points, c is the diagonal length of the *convex hull* of the two bounding boxes, α is a trade-off parameter, wt_i and ht_i are the width and the height of the bounding box $\boldsymbol{b}_i$, respectively. Differently from the standard *IoU*, the *Complete IoU* is formulated in such a way to return meaningful values, leveraging the bounding boxes geometric shapes, even when two bounding boxes are not overlapped.

Appendix B
A Better Loss for Visual-Textual Grounding

Follows the appendix of the work proposed in Chapter 5.

B.1 Model

The model follows a typical basic architecture for visual-textual grounding tasks. It is based on a two-stage approach in which, initially, a pre-trained object detector is used to extract, from a given image $\boldsymbol{I}$, a set of e bounding box proposals $\mathcal{P}_{\boldsymbol{I}} = \{\boldsymbol{p}_i\}_{i=1}^{e}$, where $\boldsymbol{p}_i \in \mathbb{R}^4$, jointly with features $H^v = \{\boldsymbol{h}_i^v\}_{i=1}^{e}$, where $\boldsymbol{h}_i^v \in \mathbb{R}^d$, where d is the number of returned features. The features represent the internal object detector activation values before the classification layers and regression layer for bounding boxes. Moreover, the model extracts the spatial features $H^s = \{\boldsymbol{h}_i^s\}_{i=1}^{e}$, where $\boldsymbol{h}_i^s \in \mathbb{R}^5$ from all the bounding boxes proposals. Specifically, the spatial features for the proposal $\boldsymbol{p}_i$ are defined as:

$$\boldsymbol{h}_i^s = \left[\frac{x1}{wt}, \frac{y1}{ht}, \frac{x2}{wt}, \frac{y2}{ht}, \frac{(x2-x1)\times(y2-y1)}{wt \times ht}\right], \tag{B.1}$$

where $(x1, y1)$ refers to the top-left bounding box corner, $(x2, y2)$ refers to the bottom-right bounding box corner, wt and ht are the width and height of the image, respectively. It is also assumed that the object detector returns, for each $\boldsymbol{p}_i$, a probability distribution $Pr_{Cls}(\boldsymbol{p}_i)$ over a set Cls of predefined classes, i.e., the probability for each class $\xi \in Cls$ that the content of the bounding box $\boldsymbol{p}_i$ belongs to ξ.

Regarding the textual features extraction, given a noun phrase $\boldsymbol{q}_j$, initially all its words $W^{\boldsymbol{q}_j} = \{w_i^{\boldsymbol{q}_j}\}_{i=1}^{l}$ are embedded in a set of vectors $E^{\boldsymbol{q}_j} = \{\boldsymbol{e}_i^{\boldsymbol{q}_j}\}_{i=1}^{l}$ where $\boldsymbol{e}_i^{\boldsymbol{q}_j} \in \mathbb{R}^w$, where

w is the size of the embedding. Then, the model applies an LSTM [105] neural network to generate from the sequence of word embeddings only one new embedding $\boldsymbol{h}_j^\star$ for each phrase $\boldsymbol{q}_j$. This textual features extraction is defined as:

$$\boldsymbol{h}_j^\star = L1\left(LSTM(E^{\boldsymbol{q}_j})\right), \tag{B.2}$$

where $\boldsymbol{h}_j^\star \in \mathbb{R}^t$ is the LSTM output of the last word in the noun phrase $\boldsymbol{q}_j$, and $L1$ is the L1 normalization function.

Once vector $\boldsymbol{h}_j^\star$ has been generated from the noun phrase $\boldsymbol{q}_j$, the model performs a multi-modal feature fusion operation in order to combine the information contained in $\boldsymbol{h}_j^\star$ with each of the bounding box proposals $\boldsymbol{h}_z^v$. For this operation, a simple function that merges the multi-modal features together is used, rather than relying on a more complex operator, such as bilinear-pooling or deep neural network architectures. The multi-modal fusion component returns the set of new vectorial representations $H^{||} = \{\boldsymbol{h}_{jz}^{||}\}_{j\in[1,\dots,m],z\in[1,\dots,e]}$, where vectors $\boldsymbol{h}_{jz}^{||}$ are defined as:

$$\boldsymbol{h}_{jz}^{||} = LR\left(\boldsymbol{W}^{||}\left(\boldsymbol{h}_j^\star \,||\, \boldsymbol{h}_z^s \,||\, L1(\boldsymbol{h}_z^v)\right) + \boldsymbol{b}^{||}\right), \tag{B.3}$$

where $||$ indicates the concatenation operator, $\boldsymbol{h}_{jz}^{||} \in \mathbb{R}^c$, LR indicates the leaky-relu activation function, $\boldsymbol{W}^{||} \in \mathbb{R}^{c\times(t+s+v)}$ is a matrix of weights, and $\boldsymbol{b}^{||} \in \mathbb{R}^c$ is a bias vector.

Finally, the model predicts the probability $\boldsymbol{P}_{jz}$ that a given noun phrase $\boldsymbol{q}_j$ is referred to a proposal bounding box $\boldsymbol{p}_z$ as:

$$\boldsymbol{P}_{jz} = \frac{\exp(\boldsymbol{W}^g \times \boldsymbol{h}_{jz}^{||} + b^g)}{\sum_{i=1}^{e} \exp\left(\boldsymbol{W}^g \times \boldsymbol{h}_{ji}^{||} + b^g\right)}, \tag{B.4}$$

where $\boldsymbol{W}^g \in \mathbb{R}^{1\times c}$ and $b^g \in \mathbb{R}$ are weights.

Indeed, the representations $\boldsymbol{h}_{jz}^{||}$ of the proposals bounding box features conditioned with the textual features can also be used to refine the proposal bounding box coordinates, which are generated by the object detector independently by the textual features. Specifically, the model does not predict new bounding box coordinates, but offsets for the coordinates defined as:

$$\boldsymbol{o}_{jz} = \boldsymbol{W}^{\mathcal{B}} \times \boldsymbol{h}_{jz}^{||} + \boldsymbol{b}^{\mathcal{B}}, \tag{B.5}$$

where $\boldsymbol{W}^{\mathcal{B}} \in \mathbb{R}^{4\times c}$ and $\boldsymbol{b}^{\mathcal{B}} \in \mathbb{R}^4$ are a matrix of weights and a bias vector, respectively. The final predicted bounding boxes coordinates are then obtained as the sum of the proposal bounding boxes coordinates with the predicted offsets.

B.2 Implementation Details

The model extracts the words' vocabulary using the SpaCy [53] framework for both datasets. Each word embedding is initialized using the GloVe [3] pre-trained weights, which the proposed model does not train, while the remaining weights are initialized with Xavier [162]. To compare objectively the experimental results with State-of-the-Art models, the same object detector adopted in [23] is used, which consists of a Faster R-CNN pre-trained object detector [66] on the Visual Genome [32] dataset that uses ResNet-101 as backbone model[1]. The features associated with each bounding box are extracted from the ResNet-101's layer *pool5_flat*. Following [23], the object detector returns for each bounding box proposal a probability distribution over 1600 classes. Other object detectors or bounding box proposals could have been applied, which would have led to further improvements, however, this research direction is not related to the aim of this work. The model adopts the normalized bounding boxes coordinates with the following representation:

$$\boldsymbol{b} = \left[\frac{x1 + x2}{2}, \frac{y1 + y2}{2}, bwt, bht \right], \tag{B.6}$$

where bwt and bht are the width and height of the bounding box, respectively.

Regarding the parameter $alpha$ in Eq. A.5, it is used the value specified in [104] which is identified by a specific formula. The proposed model comprises 10M trainable neurons and a variable number of untrained neurons (i.e., freezed) according to the dataset's word vocabulary. For Flickr30k Entities the are 6M of untrained neurons, while for ReferIt there are 2M untrained neurons.

Regarding the application of the proposed losses to the DDPN [23] model, the authors' official code of the object detector was used to extract the bounding boxes proposals with their probabilities, and then their DDPN model was re-implemented in PyTorch. Specifically, their model was implemented following the architecture and the hyper-parameters reported in their article, because the official implementation, as reported in the official repository[2], presents a slightly different architecture that leads to different results. On the re-

[1] The ResNet-101 weights were pre-trained on COCO for initialization.

[2] https://github.com/XiangChenchao/DDPN

implemented model, maintaining the same architecture and hyper-parameters, the new proposed losses where implemented.

B.3 Computational Complexity

Computational complexity among different models is a crucial aspect when comparing their performance. However, in the original research papers of the models considered in this work, the necessary information to fully understand the models' dimensions and compute power requirements is often missing. This lack of information makes it challenging to compare the model's sizes to each other directly, especially when the author's code is not made publicly available online. This problem becomes even more pronounced when considering both one-stage and two-stage approaches for solving the visual-textual grounding problem. Two-stage approaches utilize object detectors as pre-processing phase, while in one-stage approaches the object detector is included in the grounding model. Each object detector can have varying computational demands and performance according to its hyper-parameters (e.g., the number of objects to consider before the NMS component), thus adding another layer of complexity to the model comparison. However, for the sake of clarity, in the following, we report the size of the grounding models that have this information available. The proposed model comprises 10M trainable neurons and a variable number of untrained neurons (i.e., freezed) according to the dataset's word vocabulary. For Flickr30k Entities the vocabulary consist of 6M of untrained neurons, while for ReferIt there are 2M untrained neurons. DDPN [23] is composed of 24M of trainable neurons, also when adopting the new proposed losses. QRC net [74], YOLO [24], and the model proposed by F. Wu *et al.* [118] are based on the one-stage setting, and thus, it is more likely that they are composed of more trainable neurons compared to the others approaches based on the two-stage setting.

B.4 Qualitative Results

Figures B.1-B.12 report some qualitative results obtained by the proposed approach in both Flickr30k Entities and ReferIt datasets. Figures B.1, B.2, B.3, B.4, B.5, B.6 are examples of the Flickr30k Entities test set images, while Figures B.7, B.8, B.9, B.10, B.11, B.12 are examples of the ReferIt test set. It can be seen that in both the datasets, very often the predicted bounding boxes that have an intersection over union value under 0.5, are still close to the ground truths bounding boxes. Only in Figure B.5, the model predicts a bounding box for the query "one hand" that is located very far from its ground truth.

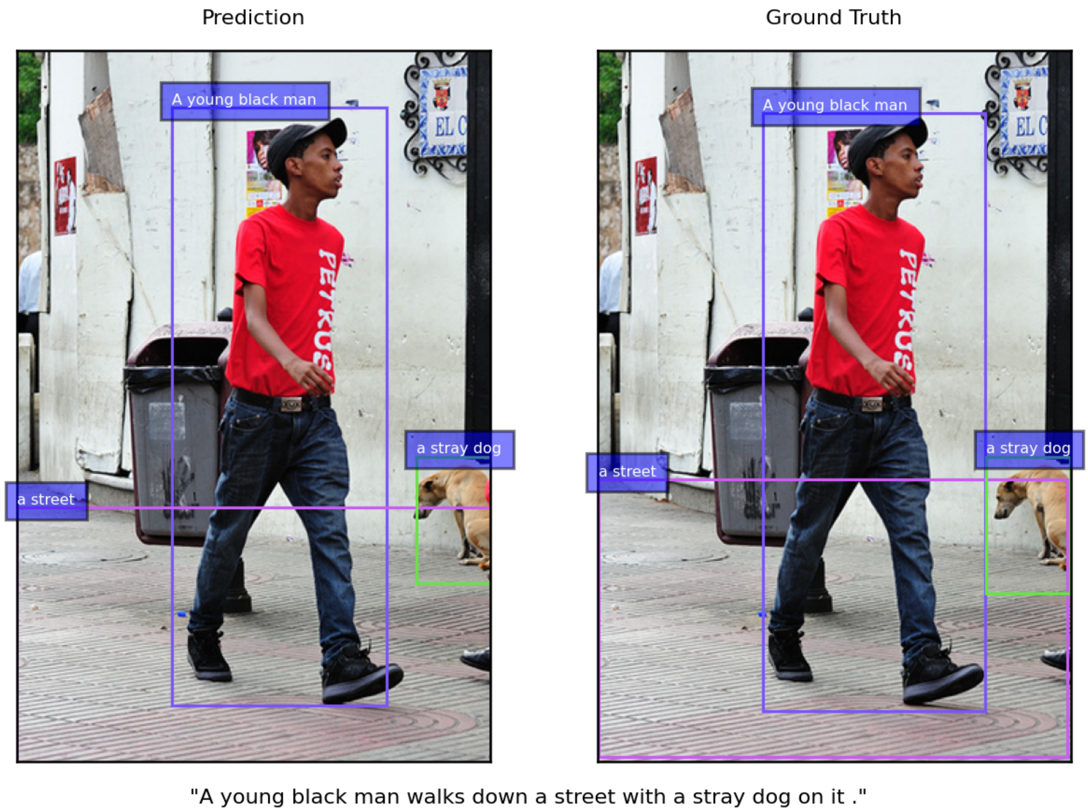


Figure B.1: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. All bounding boxes are predicted correctly.

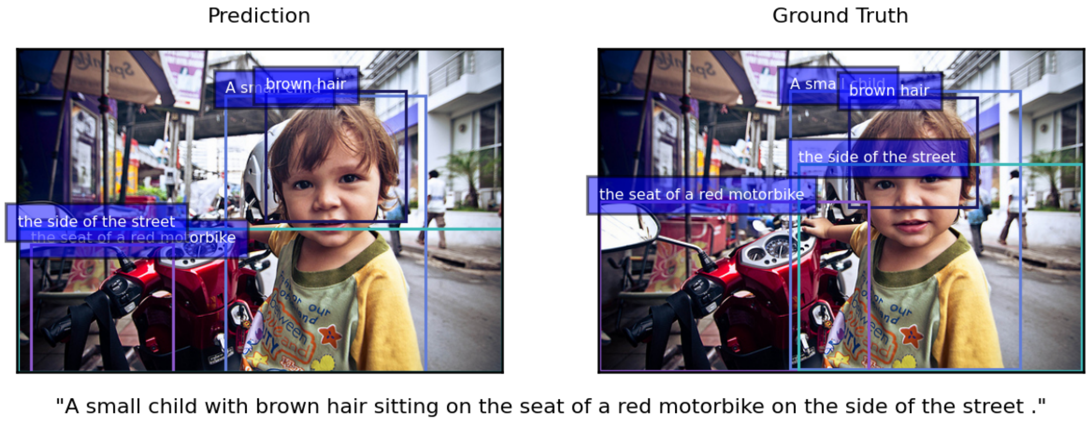

Prediction
Ground Truth
A small child
brown hair
the side of the street
the seat of a red motorbike
A small child
brown hair
the side of the street
the seat of a red motorbike
"A small child with brown hair sitting on the seat of a red motorbike on the side of the street ."

Figure B.2: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. The bounding boxes aligned with the queries “the seat of a red motorbike” and “the side of the street” present an intersection over union value with their ground truths that is lower than 0.5.

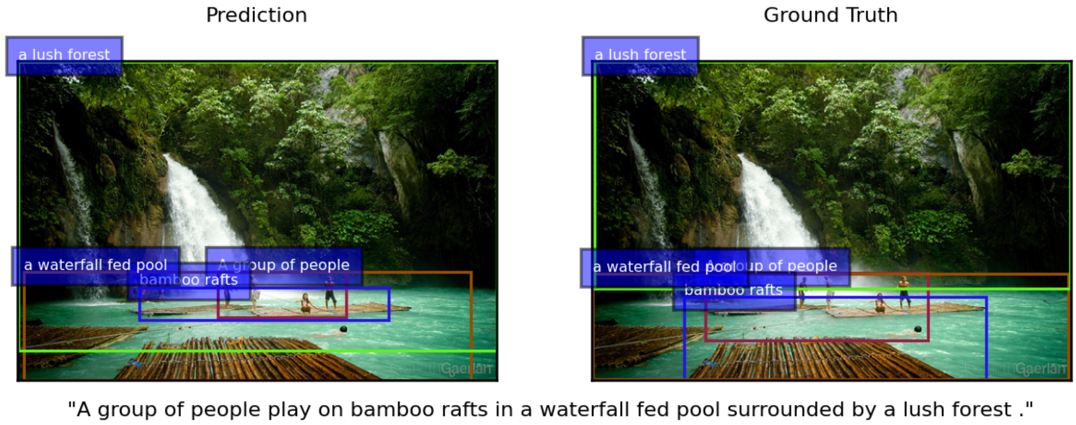


Figure B.3: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. The bounding boxes aligned with the queries “A group of people” and “bamboo rafts” present an intersection over union value with their ground truths that are lower than 0.5.

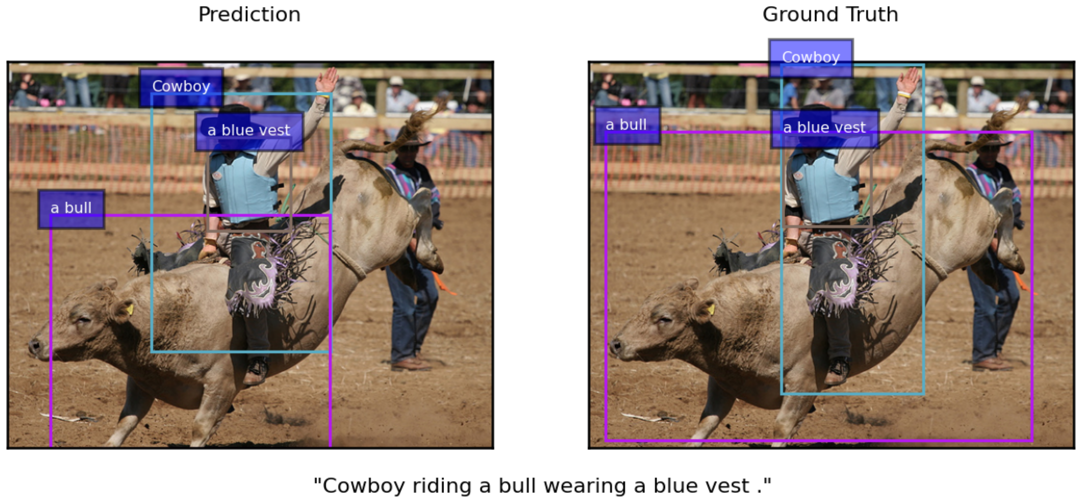


Figure B.4: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. The bounding box aligned with the query “a bull” presents an intersection over union value with its ground truth that is lower than 0.5.

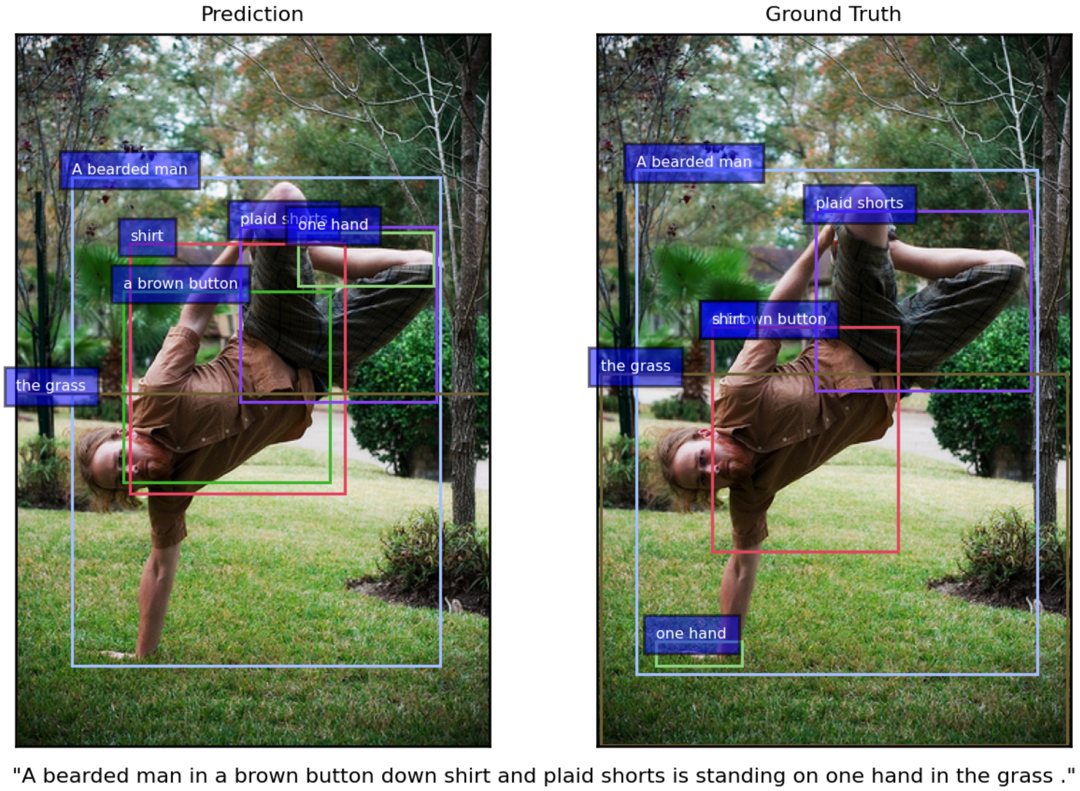


Figure B.5: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. The bounding boxes aligned with the queries “shirt” and “one hand” present an intersection over union value with their ground truths that are lower than 0.5.

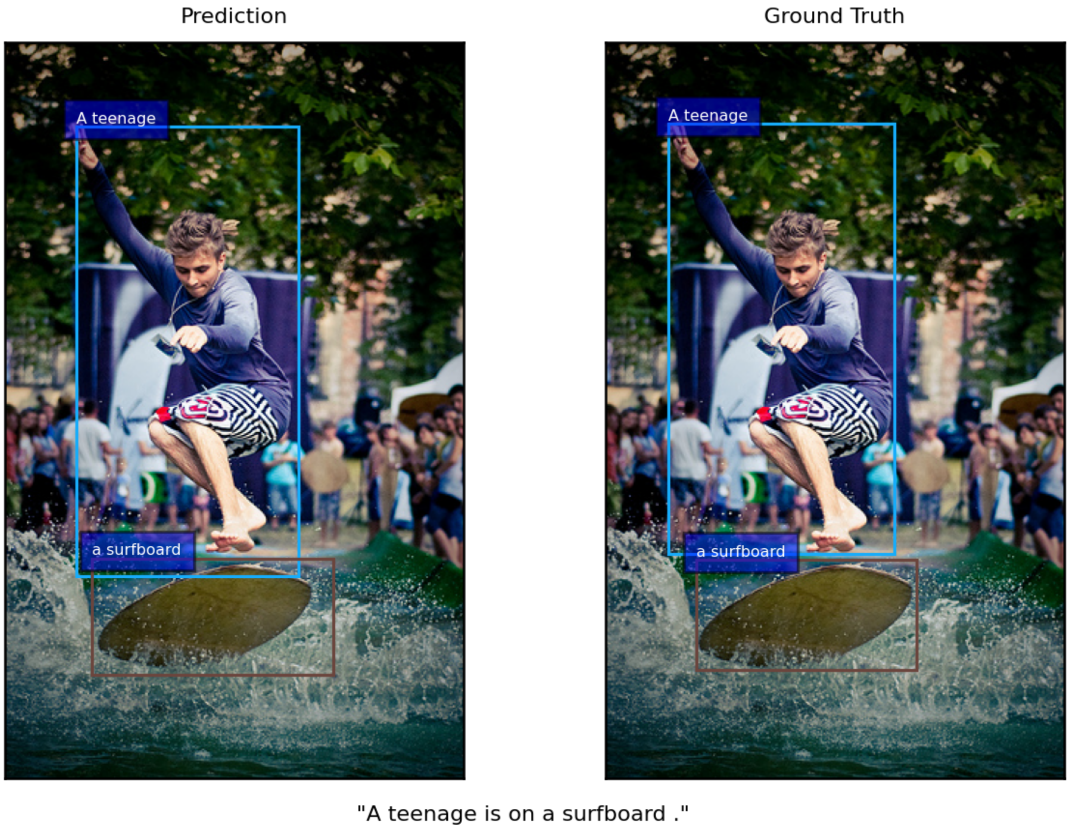


Figure B.6: Qualitative result obtained by the proposed approach on the Flickr30k Entities test set. All bounding boxes are predicted correctly.

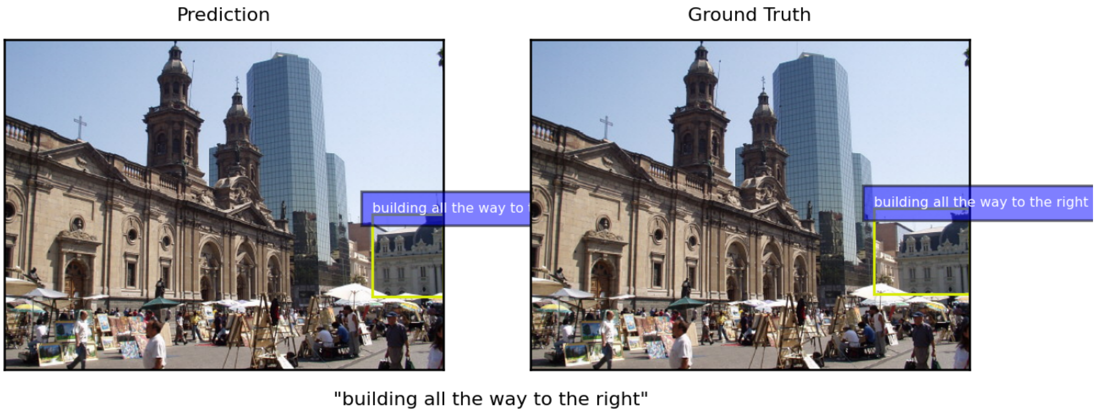

Prediction
Ground Truth
building all the way to
building all the way to the right
"building all the way to the right"

Figure B.7: Qualitative result obtained by the proposed approach on the ReferIt test set. The bounding box is predicted correctly.

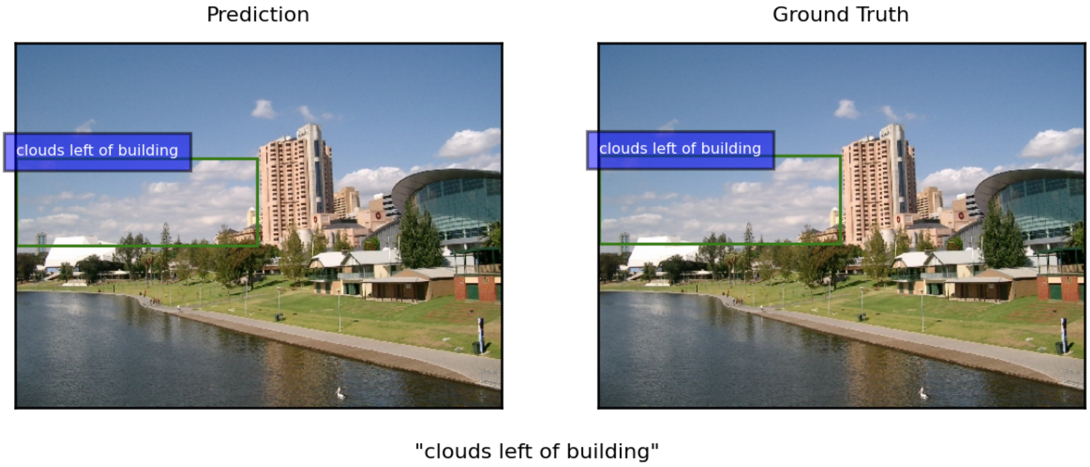


Figure B.8: Qualitative result obtained by the proposed approach on the ReferIt test set. The bounding box is predicted correctly.

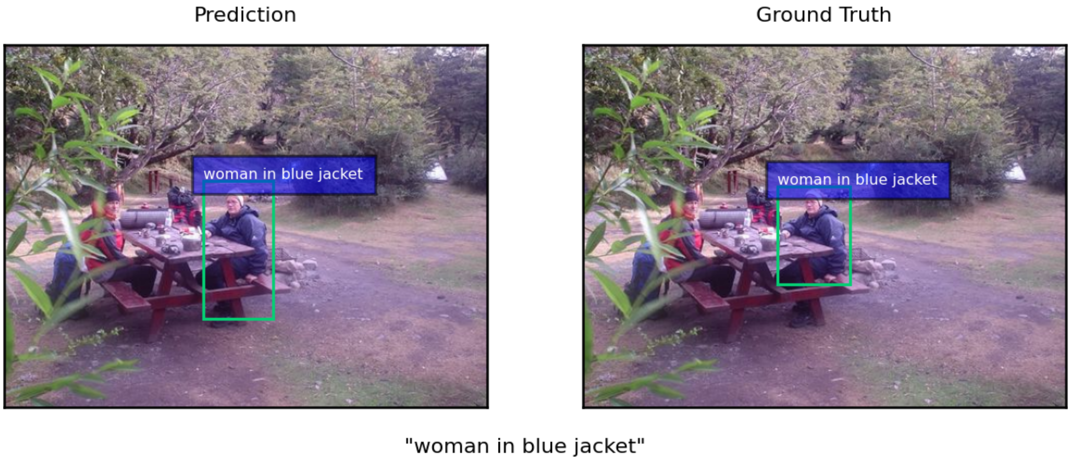


Figure B.9: Qualitative result obtained by the proposed approach on the ReferIt test set. The bounding box is predicted correctly.

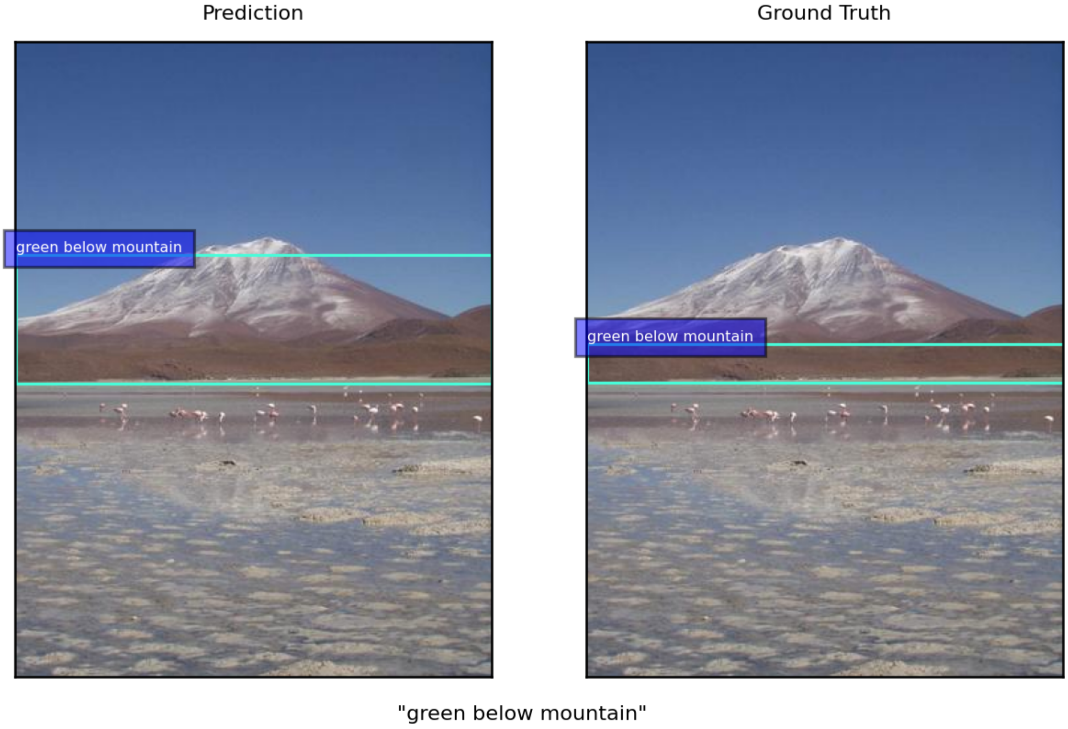


Figure B.10: Qualitative result obtained by the proposed approach on the ReferIt test set. The predicted bounding box presents an intersection over union value with the ground truth of 0.30.

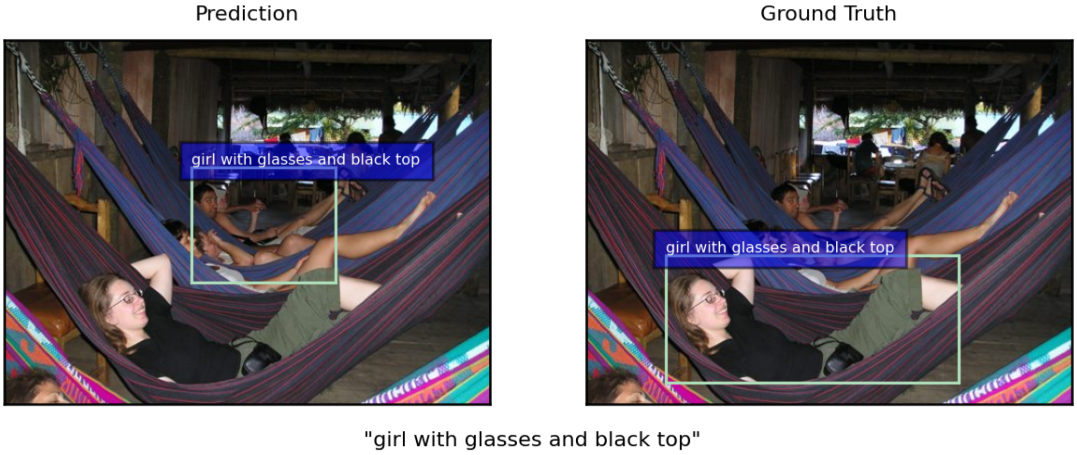

Prediction
Ground Truth
girl with glasses and black top
girl with glasses and black top
"girl with glasses and black top"

Figure B.11: Qualitative result obtained by the proposed approach on the ReferIt test set. The predicted bounding box presents an intersection over union value with the ground truth of 0.08.

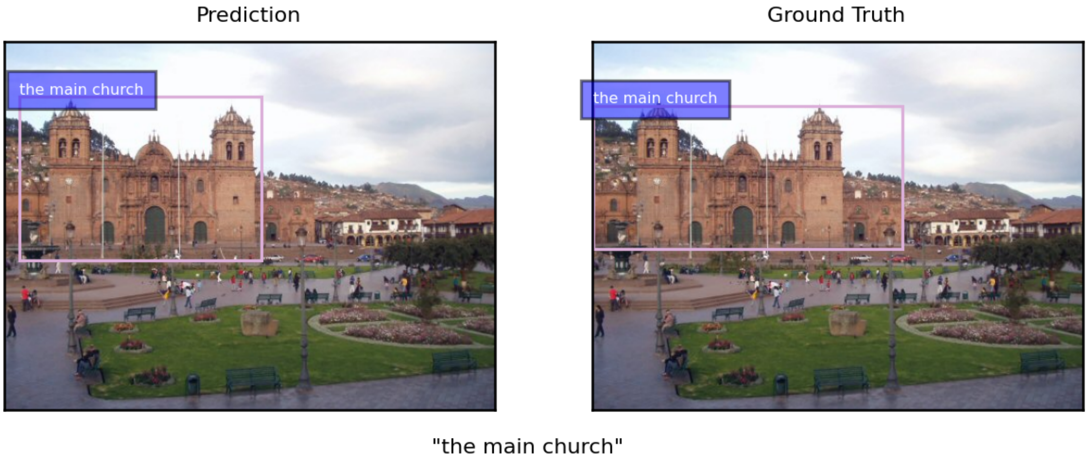


Figure B.12: Qualitative result obtained by the proposed approach on the ReferIt test set. The bounding box is predicted correctly.

Appendix C
Weakly-Supervised Visual-Textual Grounding with Semantic Prior Refinement

Follows the appendix of the work proposed in Chapter 6.

C.1 CLIP's Embeddings

This section explores the performance of the proposed model replacing representation learned by the visual and textual branch with CLIP's multimodal representations [124]. Two experiments are conducted. In the first, the proposal and query representations are replaced with CLIP's frozen embeddings. In the second, it is applied a non-linear transformation to CLIP's embeddings and trained the model. The OpenAI CLIP's implementation[1] is used with the ResNet-101 backbone. As Table C.1 shows, the proposed model does not benefit from CLIP's embedding. Such outcome may be related to how CLIP is trained, as it is not explicitly meant to work with fine-grained information such as the alignments between objects in the image and their textual references.

C.2 Computational Complexity

Computational complexity among different models is a crucial aspect when comparing their performance. However, as reported in Appendix B.3 regarding the models' computational complexity, also in this case in the original research papers of the models considered in this

[1] https://github.com/openai/CLIP

Model	Flickr30k Entities(%)	ReferIt (%)
CLIP w/o Projection	49.65	38.99
CLIP	56.89	40.99
The proposed model (without CLIP)	**62.20**	**48.03**

Table C.1: Accuracy results on Flickr30k Entities and ReferIt test sets, leveraging CLIP's multimodal embeddings. In *CLIP w/o Projection*, the visual and textual branches simply return CLIP encoded representations of proposals and queries, while *CLIP* applies a non-linear transformation to CLIP encoded representations to match the same size of the proposed model's multimodal space.

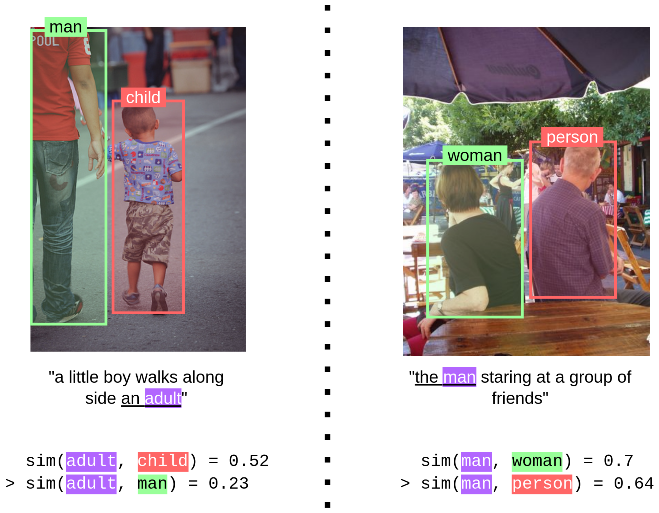

man
child
"a little boy walks along side an adult"
sim(adult, child) = 0.52
> sim(adult, man) = 0.23
woman
person
"the man staring at a group of friends"
sim(man, woman) = 0.7
> sim(man, person) = 0.64

Figure C.1: Examples of incorrectly captured similarity in word embedding space. GloVe [3] is used to compute the word embeddings. The similarity measure `sim` is the cosine similarity.

work, the necessary information to fully understand the models' dimensions and compute power requirements is often missing. However, for the sake of clarity, in the following, we report the size of the grounding models that have this information available. The SPR model comprises 241M neurons, of which 240M neurons refer to the two word vocabularies. The word vocabulary composing the Concept Branch is made of 120M untrained neurons, while the neurons of the other vocabulary are trainable. Multimodal Alignment Framework (MAF) [88] is made of 120M trainable neurons. All the considered models are two-stage approaches.

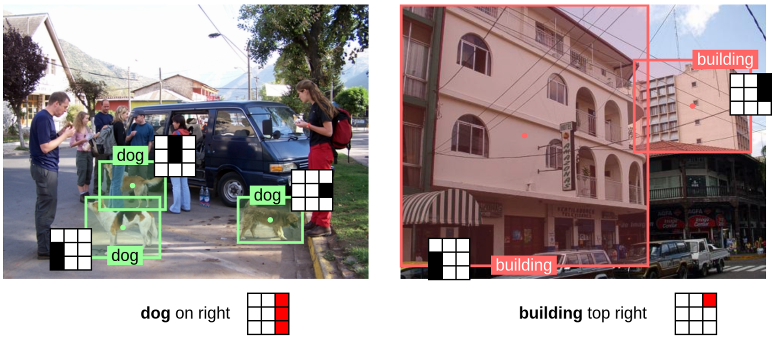


Figure C.2: Examples of how the *Concept Branch* benefits from the spatial positional information. The 3×3 square represents the positional information expressible through employed spatial knowledge, i.e. "left", "center", "right" for the horizontal axis and "top", "middle", "bottom" for the vertical axis. For the query, the spatial location is computed by a simple text search. For proposals, spatial relations are computed relating to bounding box centers. A proposal is penalized when no spatial relations are shared with the spatial location.

C.3 Limitations

The *Concept Branch* module allows the proposed model to reach state-of-the-art results also when trained on small training sets. However, although it makes the model training more robust, it is sensible to the pre-trained word embeddings which are used to initialize its weights. The accuracy of the *Concept Branch* predictions depends on the semantic information that is gathered by the pre-training embeddings between pairs of words. If the pre-training embeddings are unable to gather sufficient semantic information, then the accuracy of the *Concept Branch* predictions will decrease. Figure C.1 provides two examples of incorrectly captured similarity in word embeddings. On the left side of the figure, the words "adult" and "child" have a higher similarity compared to "adult" and "man". As a result, the *Concept Branch* suggests the wrong alignment. On the right side of the figure, the correct bounding box is labeled as "person", although it may not be the most precise. However, the *Concept Branch* suggests a different alignment due to the higher similarity between the words "man" and "woman".

One issue that instead arises with object detector classification is when the bounding boxes are classified with the same label, which can lead to ambiguity in the predictions made by the *Concept Branch*. This occurs because the similarity between the query and the labels is identical. In such situations, the model leverages positional relative knowledge to address the problem by directing attention to candidates that share at least one positional reference with the query. Figure C.2 illustrates how this straightforward positional heuristic can be employed to overcome this challenge.

Finally, since the proposed model is based on the proposals predicted by a pre-trained object detector, it suffers from the errors made in the prediction of the bounding boxes and their classes. In fact, to associate an object in the image with its textual reference, the object must first be localized by the object detector.

Appendix D
Cleaner categories improve object detection and visual-textual grounding

Follows the appendix of the work proposed in Section 7.1.

D.1 Frequencies by Categories

Section 7.1 introduced both the set of clean and random categories deriving from the original ones. The original label set is defined by 1600 categories, while both the new clean and the random sets are defined by 878 categories. Figure D.1 shows frequencies of objects appearing in the Visual Genome training split, where objects are either labeled according to the original label set (in blue), the new cleaned label set (in orange), or the random label set (in brown). The new label sets lead mostly to the removal of many low-frequency categories in the long tail, rather than creating new very frequent categories. Surprisingly, the random procedure that generated the random label set also removed the long tail of low-frequencies categories.

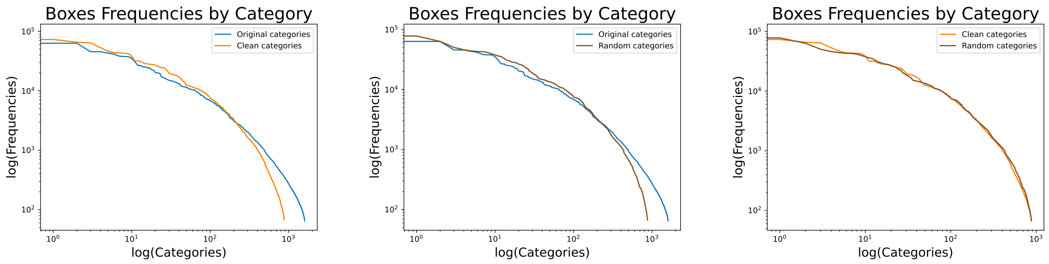


Figure D.1: LogLog plots of objects frequencies for each category. The frequencies are calculated on the training set annotations. The distribution of the original categories is in blue, the new categories are in orange, and the random categories are in brown. The cleaning process did not generate high-frequency categories and at the same time removed many low-frequency categories for both cleaner and random label sets.

D.2 Prediction Confidence

In Figure D.2 it is reported the KDE plots for the probability values of the argmax category predicted by the original, clean, and random label sets.

The BUA detector trained on the cleaned categories produces more high confidence predictions than a detector trained on the original noisy categories. Closer inspection shows that this difference is due to higher confidence when predicting objects in the new merged clean categories. However, this is not the case for BUA trained on random categories, which presents the same confidence as the model trained on the original categories.

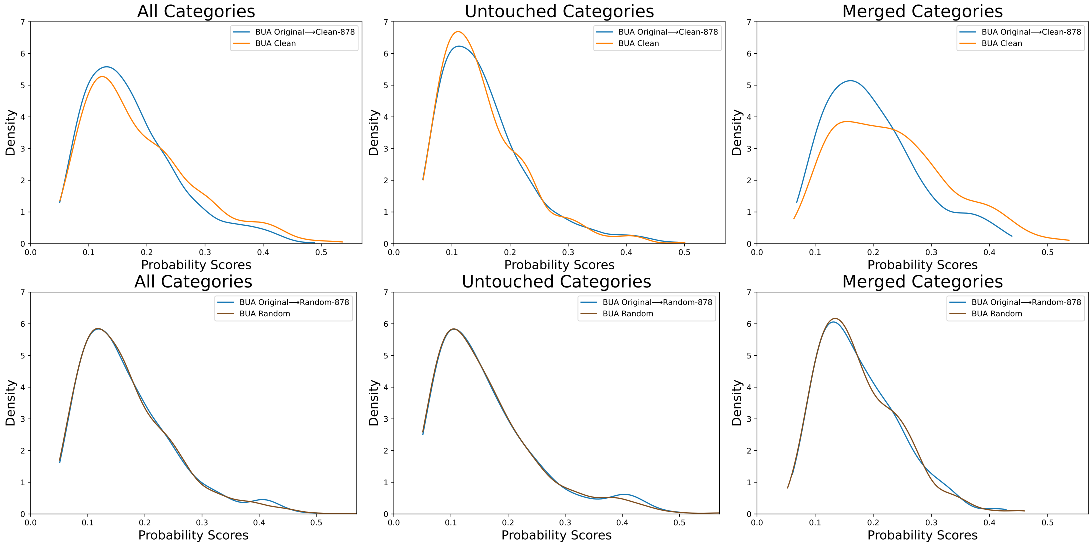


Figure D.2: KDE plots for the probability values of the argmax category predicted by the model. The plots on the left consider all the categories, the plots in the center consider just the categories that not merge during the cleanup process (i.e., "Untouched"), and the last plots on the right consider only the merged categories. Overall, the cleaned categories lead to higher confidence values than the original categories, while there is no difference between original and random categories.

D.3 Nearest Neighbors Analysis on Random Labels

This section performs the nearest neighbors analysis on the random labels focusing on the "Merged", "Untouched", and "All" categories. Table D.1 reports the results of this analysis, considering features extracted with different threshold values (i.e., 0.05 and 0.2) and considering either all features or only features from different images ("Filtered Neighbors"). This step removes features that might be from highly overlapping regions of the same image.

Th.	K	Categories	All Neighbors (%)		Filtered Neighbors (%)	
			Original	Random	Original	Random
0.05	1	All	12.15±12.25	12.36±11.15	37.32±15.07	37.83±12.32
0.05	1	Untouched	10.06±11.91	10.32±12.13	35.81±13.91	36.33±14.03
0.05	1	Merged	13.16±12.29	11.35±10.50	38.05±15.55	38.56±12.90
0.05	5	All	24.33±24.38	24.91±12.01	34.16±13.78	34.68±12.24
0.05	5	Untouched	22.66±12.60	23.12±12.61	33.09±12.88	33.54±12.78
0.05	5	Merged	25.13±13.66	25.77±11.61	34.68±14.16	35.23±11.93
0.05	10	All	27.76±13.23	28.37±11.87	32.91±13.71	33.48±12.19
0.05	10	Untouched	26.40±12.34	26.98±12.04	31.89±12.78	32.42±12.71
0.05	10	Merged	28.42±13.60	29.04±11.55	33.39±14.12	33.99±11.89
0.2	1	All	51.02±22.74	51.88±20.91	69.22±18.99	70.03±16.76
0.2	1	Untouched	45.05±21.50	46.30±21.68	65.93±17.39	66.70±17.10
0.2	1	Merged	53.84±22.73	54.70±19.93	70.98±19.37	71.72±16.33
0.2	5	All	60.40±19.75	61.47±17.84	65.12±19.68	66.12±17.54
0.2	5	Untouched	56.60±18.22	57.61±17.99	61.87±18.10	62.75±17.67
0.2	5	Merged	62.33±20.20	63.42±17.45	66.79±20.22	67.82±17.23
0.2	10	All	60.55±20.18	61.71±18.20	62.95±20.43	64.05±18.34
0.2	10	Untouched	57.05±18.44	58.14±18.24	59.76±18.69	60.69±18.39
0.2	10	Merged	62.31±20.78	63.51±17.91	64.56±21.06	65.75±18.07

Table D.1: Proportion of K-nearest neighbors that share the same predicted category, comparing models trained using the original versus random categories (cf. Table 7.3). The random features present small improvements over the original features, suggesting that there is a small advantage in training with fewer labels; however clean labels help more.

The random features present results very similar to those obtained with the original features, but with a small improvement. In other words, there is an advantage to training on fewer labels overall. However, the improvement given by clean labels is much greater than that obtained with the random labels, strengthening the importance of training BUA with clean categories.

Appendix E

Object Search by a Concept-Conditioned Object Detector

Follows the appendix of the work proposed in Section 7.2.

E.1 Dataset Statistics

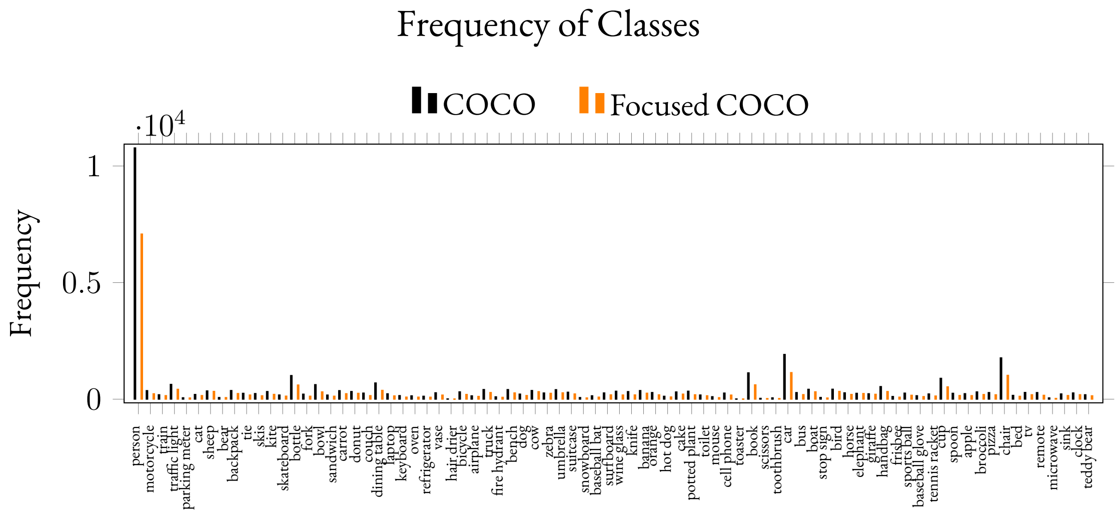


Figure E.1: Frequencies of the classes appearing in the test set of both COCO and Focused COCO datasets. The COCO test set refers to the original COCO validation set.

The Visual Genome [32] dataset consists of 108077 images with an average width of 500 pixels. Each bounding box is classified with a class belonging to a set of 1600 categories extracted in the work of Ranjay *et al.* [66]. Every split of data is available online with its ground

Statistic	COCO	Visual Genome		Focused COCO	Focused Visual Genome	
	Test	Valid	Test	Test	Valid	Test
Number of images	5000	5000	5000	5000	5000	5000
Average number of bounding boxes	7.4	30.4	30.2	4.8	16.8	16.7
Max number of bounding boxes	63	154	118	56	116	99
Min number of bounding boxes	0	0	0	0	0	0
Average number of unique classes	2.9	16.5	16.4	1.9	9.1	8.9
Max number of unique classes	14	43	43	12	35	34
Min number of unique classes	0	0	0	0	0	0
Average N. of concepts	7.4	30.4	30.2	4.8	16.4	16.3
Max N. of concepts	63	154	118	56	115	96
Min N. of concepts	0	0	0	0	0	0

Table E.1: Statistics per image about the datasets augmented with concepts adopted in this work. The COCO test set refers to the original COCO validation set. The training set examples are generated at "run-time" during training.

truth annotations. Hence, on this dataset, the online splits are adopted to generate the sets used for training, validating, and testing the proposed approaches. The 1600 categories are derived from textual phrases, and even if they are the results of a sophisticated cleaning process as reported in [66], they are still noisy and sometimes ambiguous. For example, there are categories representing the same meaning but written in different ways, categories representing single and plurals of the same concepts such as "MAN" and "MEN", ambiguous categories such as "LADY", and also classes that represent attributes as "YELLOW".

The test set of both COCO and Focused COCO datasets are not publicly available online, and as explained in Section 7.2.4.1, the original validation set was used for testing.

Table E.1 reports the statistics of the dataset considered in this work. Figure E.1 reports the frequencies of the classes appearing in the test set of both COCO and Focused COCO datasets. As the results show, the Focused COCO derived by the original test set resembles the distribution of the COCO original test set.

E.2 Model Selection

Given the large computational power required for training the object detectors, the search for the best hyper-parameters was performed only on the COCO dataset. Thus, the best hyper-parameters selected on COCO are adopted "as-is" for training the object detectors on the Visual Genome dataset. The model hyper-parameters tuning is performed by training on the train set and validating on the validation set (the one randomly sampled from the training set. See Section 7.2.4.1 for more details.). The evaluation results presented in this work are always obtained on the test set (i.e., the original validation set).

All models are trained for 90K iterations and are then tested on the validation set. Hyperparameters related to concepts are tuned using RetinaNet [65], with ResNet-50 and Feature Pyramid Network (FPN) [157], on the COCO dataset.

Regarding the *Fusion Block*, three approaches o fuse the concept embeddings with the visual features are tried. In particular, it is tried to add, multiply and concatenate the concept embeddings with the visual features. The best AP results were obtained by adopting the concatenation approach. The best learning rate to use during training is searched among the following values: $[0.01, 0.001, 0.0001, 0.00005]$. With DynamicHead architectures, the best results were achieved with a value of 0.0001, while with RetinaNet architectures, the best results were achieved with a learning rate value of 0.01. Also, the addition of more expressiveness to the *Concept Set Encoding* network was investigated, but the best results were obtained with the configuration reported in Appendix E.3.

E.3 Implementation Details

For model training, all ResNet [72] backbones are initialized with the pre-trained ImageNet [163] weights. The Swin-Tiny backbone is initialized with the weights provided by the authors[1]. As concept embedding, the 150-dimensional Holographic [164] embeddings[2] trained on WordNet for 500 epochs is adopted. These weights are frozen during model training. The batch size is fixed to 16 examples in training all models. The *Concept Set Encoding* module employs a Deep Sets [165] network. Each 150-dimensional concept embedding is mapped to a new 256-dimensional representation using a multilayer perceptron with two layers and ReLU activation functions. The first layer has a dimension of 150 neurons, while the second layer has a dimension of 256 neurons. Finally, all the concepts' representations are summed and transformed into a new representation with a multilayer perceptron with two 256-dimensional layers and ReLU activation functions. The *Fusion Block* concatenates the embedding of the concepts, in output from the *Concept Set Encoding*, to the visual features in output from the model *Backbone*. Each object detector category is mapped[3] to its corresponding WordNet synset using the Python NLTK[4] package. When NLTK failed to find the concept associated with some categories, the linking was done manually with the synset that most represented the category meaning. Were not explicitly indicated, all the concepts

[1] https://github.com/microsoft/DynamicHead

[2] https://github.com/drigoni/WordNet_Embeddings

[3] This implements the f function.

[4] https://www.nltk.org/

Meta Architecture	Fusion Strategy	Backbone		Concept Vocab.	Params on COCO		Params on Visual Genome	
		Type	Params		Head	Total	Head	Total
RetinaNet	/	ResNet-50	31.5M	0	6.5M	37.9M	38.0M	69.4M
Concept RetinaNet	Concat	ResNet-50	31.5M	12.4M	22.4M	66.5M	85.4M	129.3M
RetinaNet	/	ResNet-101	50.4M	0	6.5M	56.9M	38.0M	88.4M
Concept RetinaNet	Concat	ResNet-101	50.4M	12.4M	22.4M	85.4M	85.4M	148.2M
DynamicHead	/	ResNet-50	38.8M	0	2.4M	41.2M	2.8M	41.6M
Concept DynamicHead	Concat	ResNet-50	38.8M	12.4M	9.5M	68.8M	10.3M	61.5M
DynamicHead	/	ResNet-101	57.8M	0	2.4M	60.2M	2.8M	60.5M
Concept DynamicHead	Addition	ResNet-101	57.8M	12.4M	2.4M	72.6M	2.8M	73.0M
Concept DynamicHead	Concat	ResNet-101	57.8M	12.4M	9.5M	79.7M	10.3M	80.5M
DynamicHead	/	Swin-Tiny	42.3M	0	2.4M	44.7M	2.8M	45.1M
Concept DynamicHead	Concat	Swin-Tiny	42.3M	12.4M	9.5M	64.3M	10.3M	65.0M

Table E.2: Number of parameters for each model. The number of parameters of the object detector heads varies according to the number of classes to detect. The models conditioned with the concepts are highlighted with the dove gray color. *Fusion Strategy* refers to the function applied to fuse the visual and concept information (more details in Appendix E.4).

sampling procedures are done at a maximum depth of $d = 1$. The proposed models are implemented using the Detectron2 framework[5]. All the experiments were performed in a distributed parallel system using several A100 40GB GPUs[6]. Table E.2 presents the number of parameters composing each model. In particular, the table reports the number of parameters forming the backbone, the size of the concept vocabulary, and the number of parameters composing the head of the model. The head of the model is in charge of locating and classifying the objects in the image, and for this reason, its dimension depends on the number of classes to predict. In other words, the size of the model's head changes according to the dataset. *Fusion Strategy* refers to the function applied to fuse the visual and concept information. More details are reported in Appendix E.4.

E.4 Further Analysis on Fusion Block

The block that fuses information from the visual modality with information from the concepts modality (i.e., *Fusion Block*) plays a key role in the construction of the conditioned object detector. The best results were obtained with the concatenation (i.e., "Concat.") of the features, which implies a larger *Object Detector Head*'s input (i.e., slightly more neurons) that could explain the improvement in the object detector capabilities.

To discern if that is the case, in the following, it is reported the results obtained using the "Addition" strategy, which sums the visual and concept features without increasing the size of the *Object Detector Head*, i.e., the same number of parameters. Appendix E.3 reports the number of neurons constituting each model.

Table E.3 presents the results obtained following the evaluation setting of Section 7.2.4.3, while Table E.4 presents the results obtained following the setting of Section 7.2.4.4. For these experiments, it is considered the architecture DynamicHead with ResNet-101 as the backbone.

In both settings, it can be observed that the Concept-Conditioned DynamicHead models adopting the "Addition" fusion strategy performs much better than the standard DynamicHead models. On the other hand, the "Addition" strategy is not as competitive as the "Concatenation" strategy.

In conclusion, these results demonstrate that, albeit more neurons help the object detector performance, the major results' improvement is given by the concepts in input.

[5] https://github.com/facebookresearch/detectron2

[6] Code available at: https://github.com/drigoni/Concept-Conditioned-Object-Detector

Meta Architecture	Backbone	COCO (AP%)					Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
DynamicHead	/	44.1	26.7	47.7	57.2	61.6	5.7	2.9	6.1	8.2	10.1
Concept DynamicHead	Addition	49.7	32.3	53.7	64.2	70.7	8.4	4.5	8.7	11.4	15.7
Concept DynamicHead	Concat.	50.2	33.9	54.3	64.8	71.6	9.6	5.2	9.8	13.1	18.0

Table E.3: Object detection results considering different multimodality fusion strategies. The concept-conditioned object detector searches for all the objects depicted in the image. These results are obtained using the proposed concept-conditioned DynamicHead model with ResNet-101 and $d = 1$.

Meta Architecture	Backbone	Focused COCO (AP%)					Focused Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
DynamicHead	/	49.2	32.0	52.6	62.6	69.3	10.9	5.4	10.5	14.6	19.3
Concept DynamicHead	Addition	52.1	35.2	54.9	65.7	73.5	12.7	7.1	12.2	16.7	23.2
Concept DynamicHead	Concat.	52.2	35.9	55.3	66.6	73.7	13.7	7.7	13.3	18.3	25.1

Table E.4: Object detection results considering different multimodality fusion strategies. The concept-conditioned object detector searches for a subset of objects depicted in the image. These results are obtained using the proposed concept-conditioned DynamicHead model with ResNet-101 and $d = 1$.

E.5 Comparing Standard and Concept-Conditioned Object Detectors with Filtering

Bear in mind that the experimental setting adopted in this section **does not** reflect the scope of the work presented in Section 7.2. Indeed, the scope of the work introduced in Section 7.2 is to highlight the importance of conditioning object detectors for searching only a subset of objects appearing in a stream of images. While, this section focuses on the specific case in which object detectors, coupled with the *Post-processing Selection* component, are deployed to search all the objects depicted in the image, i.e., $G(\boldsymbol{I})$. In other words, assuming that it is available at priori the set of concepts related to all the objects in the image, one would like to answer the following question: *do object detectors get better at detecting all objects when coupled with the post-processing algorithm?*

Table E.5 presents the results obtained in this new setting. In both datasets, the concept-conditioned object detectors perform always better than standard object detectors. More specifically, the average improvements obtained with Concept DynamicHead compared to DynamicHead are higher than those obtained with Concept RetinaNet compared to RetiNanet.

Overall, also in this setting, the proposed concept-conditioned models using the user's intent always perform better than standard object detectors.

E.6 Object Detection Results by Class

Given the proven improvement due to the use of concepts, this section analyses the results obtained by class. The goal is to see if some concepts influence some classes more than others.

Figure E.2 presents the AP metric values obtained per class in the Focused COCO dataset by DynamicHead with Swin-Tiny, coupled with the *Post-processing Selection* component. In other words, the models search for a subset of objects in the image. The figure shows that the Concept-Conditioned DynamicHead model obtains higher results than the standard DynamicHead model in most classes. However, it presents lower results only in a small number of classes, such as "HAIR DRIER" and "KNIFE". Future works will investigate these classes more in detail.

Meta Architecture	Backbone	COCO (AP%)					Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
RetinaNet	ResNet-50	39.5	25.2	43.3	51.1	61.8	6.2	3.2	6.0	7.9	12.2
Concept RetinaNet	ResNet-50	40.4	27.6	44.3	52.0	63.4	6.3	3.4	6.3	7.8	13.0
RetinaNet	ResNet-101	41.8	26.1	46.3	54.1	64.1	6.5	3.3	6.5	8.6	13.0
Concept RetinaNet	ResNet-101	42.3	27.2	46.7	54.0	65.4	6.6	3.5	6.4	8.4	13.2
DynamicHead	ResNet-50	47.6	30.7	51.0	61.5	67.6	9.7	4.5	9.3	12.5	17.5
Concept DynamicHead	ResNet-50	50.3	33.6	54.1	65.2	71.7	11.4	6.0	11.0	14.8	21.4
DynamicHead	ResNet-101	47.7	30.2	51.6	61.5	67.5	9.8	4.4	9.5	12.9	17.6
Concept DynamicHead	ResNet-101	50.4	34.1	54.5	64.9	71.8	11.7	5.9	11.3	15.3	21.7
DynamicHead	Swin-Tiny	52.7	36.0	56.4	66.7	72.9	11.3	5.4	10.8	15.0	19.9
Concept DynamicHead	Swin-Tiny	54.9	38.2	58.7	68.3	76.3	12.6	6.8	12.6	16.5	23.5

Table E.5: Object detection results on COCO and Visual Genome datasets with $d = 1$. All the object detectors are coupled with the *Post-processing Selection* component and aim to detect all the objects depicted in the image.

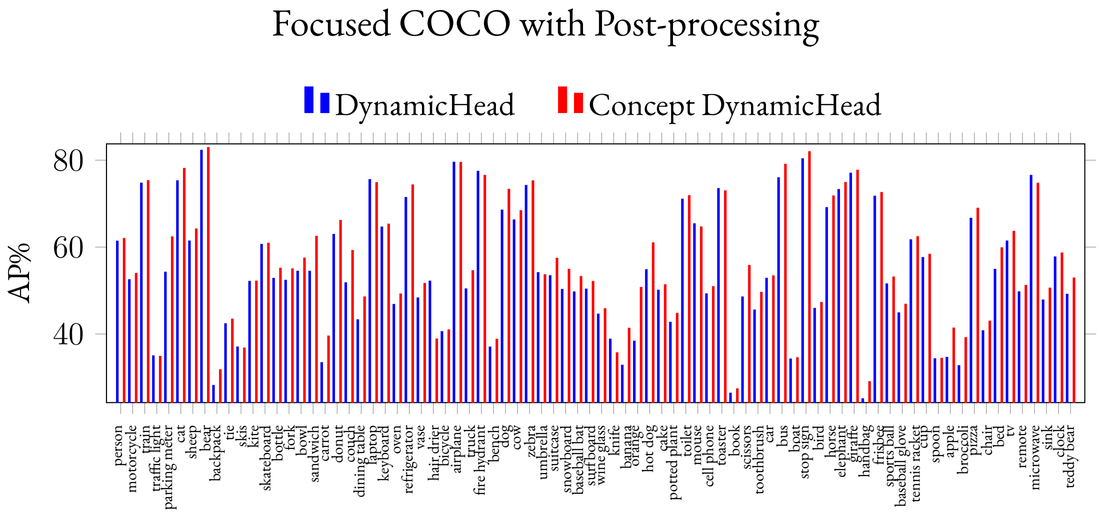


Figure E.2: Results obtained on the Focused COCO test set for each category using the Post-processing Selection. The bars in blue refer to the values obtained with the model DynamicHead and Swin-Tiny as the backbone. The bars in red refer to the values obtained with the model Concept DynamicHead and Swin-Tiny as the backbone. *AP* refers to the *AP* metric.

E.7 Qualitative Results

Figure E.3 presents some qualitative examples predicted with the proposed model and a standard object detector. It is highlighted with **red lines** the ground truth bounding boxes and with **light blue dashed lines** the bounding boxes predicted by the model. The column on the left reports the prediction of DynamicHead, while the center and right columns present the predictions obtained with the proposed Concept DynamicHead given the concepts highlighted under the images. The left and center columns use annotations from the COCO dataset (i.e., all the image ground truth), while the right column uses the Focused COCO annotations (i.e., it focuses the detection on a subset of objects).

The standard object detector focuses its attention on all the objects in the images and sometimes is not able to detect the most important bounding boxes, like the “KEYBOARD” in the first row and the “COW” in the last row. On the contrary, the concept-conditioned object detector focuses its attention only on those bounding boxes that express concepts in input, improving the bounding boxes’ detection performance and decreasing the number of detected boxes when compared to standard object detectors. On the right column, it is highlighted the use case of the proposed model, which is when the object detector is used to focus the detection only on a subset of objects depicted in the image. An interesting mismatch between concepts and object detector classes is given by the second image in the right column.

Given the concept “male.n.01” the object detector focuses its detection on the bounding box depicting the woman and classifies it as “PERSON”. Clearly, the concept in input was focusing only on males, but the object detector class that most approximate that concept is “PERSON”, as ”MALE” is not a COCO class. In fact, also the ground truth bounding box is classified as “PERSON”.

E.8 Comparing the Proposed Model to Visual-Textual Grounding Model

In this section, it is elaborated more on the comparison of the proposed model to a visual-textual grounding (i.e., referring expression) model.

As presented in Section 7.2.5, the referring expression task presents several points of difference from the approach proposed in Section 7.2, which are summarized below. First of all, the user’s intent needs to be represented as a textual phrase, while in the proposed approach, the user’s intent is expressed with one or more WordNet [33] concepts. Secondly, following the current State-of-the-Art, referring expression models predict only the bounding box that best matches the textual phrase in the output. For this reason, when the user’s intent concerns multiple different objects depicted in the image, multiple independent queries should be performed to retrieve all objects of interest. In addition, when the user’s intent concerns multiple objects of the same type, the referring expression approach is no longer suitable. Lastly, for training, referring expression models need to use detailed datasets comprising images, boxes coordinate, textual phrases, and the ground truth of the corresponding bounding box in the image for each phrase. These annotations are difficult to collect, so the referring expression datasets contain fewer examples than those of detection.

Still, given all these differences, Fornoni *et al.* [158] performed a comparison between an SSD [29] object detector coupled with ResNet-101 and a One-Stage BERT referring expression recognition model [24]. In particular, the SSD model’s results were filtered according to the class expressed by the query in input, as it is done in the baseline proposed in Section 7.2. To summarize, Fornoni *et al.* verified that the referring expression model has poor generalization ability and underperforms a simple post-processing baseline. More details are reported in Section 4.4 *ReferIt and post-processing baselines for SLD* of the Fornoni *et al.* main manuscript [158].

These results further motivate the necessity of conditioning object detectors with prior information and thus support the idea presented in this proposed work.

Statistic	COCO	Visual Genome		Focused COCO	Focused Visual Genome	
	Test	Valid	Test	Test	Valid	Test
Number of images	5000	5000	5000	5000	5000	5000
Average number of bounding boxes	7.4	30.4	30.2	4.8	16.8	16.7
Max number of bounding boxes	63	154	118	56	116	99
Min number of bounding boxes	0	0	0	0	0	0
Average number of unique classes	2.9	16.5	16.4	1.9	9.1	8.9
Max number of unique classes	14	43	43	12	35	34
Min number of unique classes	0	0	0	0	0	0
Average number of concepts	2.9	16.3	16.2	1.9	8.7	8.5
Max number of concepts	14	42	43	12	34	34
Min number of concepts	0	0	0	0	0	0

Table E.6: Statistics per image about the datasets generated with the new sampling strategy. Note that only the statistics of the concepts have changed. The COCO test set refers to the original COCO validation set. The training set examples are generated at "run-time" during training.

E.9 Concept Sampling Impact

During the model training and the creation of the new datasets with concepts, two sampling processes take place. The former aims to reduce the exponential number of examples deriving by the powerset approach $\xi(\boldsymbol{I})$ (i.e., $\hat{\xi}(\boldsymbol{I})$), while the latter aims to sample S_d to obtain a reasonable amount of concepts. More details regarding the model training are reported in Section 7.2.3.2.

This section reports the results obtained by changing the sampling process applied to S_d to obtain $\hat{S_d}$[7]. Instead of sampling one concept for each object to search in the image (as done in Section 7.2), the model's performances are analyzed when in input is provided a concept for each type of object to search in the image. This is a more generic setting than before, as the prior information concerns only the types of objects to search for and not the number of occurrences of the same object in the image. For example, given Figure 7.3 of Section 7.2, in this new setting, two concepts are provided in input: one concept for the object labeled as "BOWL" and **one** sampled concept associated with the objects labeled as "CAT". Following the same example, all the experiments previously performed (i.e., the sampling strategy used in Section 7.2) provided as input three concepts, one for "BOWL" and two sampled concepts for "CAT", i.e., one for each cat appearing in the image.

Of course, the new sampling strategy is also adopted for generating new test sets. More in detail, the new "Focused" datasets adopted in this section are built starting from the "Focused" datasets presented in Section 7.2, where only one concept is kept for each type of

[7] Only Concept-Conditioned Object Detectors need new training.

object. This implies that new datasets have the same ground truth as the starting datasets and that the only differences are in the concepts. Going back to the previous example, given the two concepts related to the two cats appearing in the image, only one concept is sampled and adopted as input for searching for both cats in the picture. Table E.6 reports the statistics of the new datasets generated with the new sampling strategy. It is evident, that only the statistics about the concepts have varied.

E.9.1 Comparing Standard and Concept-Conditioned Object Detectors before Filtering

Table E.7 presents the results obtained by the object detectors when they are deployed for searching all the objects contained in COCO and in Visual Genome datasets, as done in Section 7.2.4.3. As the results show, there is the same trend highlighted in Table 7.6 of Section 7.2: when the object detector is conditioned with concepts, it improves the ability to localize the objects in the image. On COCO, the larger AP improvement (5.2%) is given by Concept DynamicHead (49.3%) over DynamicHead (44.1%), both with ResNet-101. Even on Visual Genome, the same architecture and backbone give the best improvements (3.5%).

Table E.8 highlights the impact of employing different depth values on the proposed conditioned models adopting the new sampling strategy. The results were obtained with RetinaNet, using ResNet-50 as the backbone, on the COCO test set. As can be seen from the table, the best AP result is obtained with a depth value of 0, and there is no abrupt deterioration in the results, increasing the depth value from 0 to 4. More in detail, from the depth value of 0 to 1, the biggest deterioration in the AP metric amounts to 1%, although from the depth value of 1 to 4, the deterioration amounts to 0.5% In conclusion, these results suggest that in the COCO dataset, it is possible to generalize the model to the use of 7274 different WordNet concepts trading off some of the effectiveness of the model.

E.9.2 Searching for a Subset of Objects

In this section, concept-conditioned object detectors are compared against standard object detectors, both *coupled* with the *Post-processing Selection* component, to search for just a *subset* of objects depicted in the images and consistent with the input concepts. This evaluation setting complies with that of Section 7.2.4.4. Both Focused COCO and Focused Visual Genome are new versions of datasets generated following the new sampling strategy.

This table shows that concept-conditioned models outperform standard object detectors

Meta Architecture	Backbone	COCO (AP%)					Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
RetinaNet	ResNet-50	36.4	20.7	40.0	47.5	56.0	3.5	2.1	3.8	5.0	7.0
Concept RetinaNet	ResNet-50	38.4	24.8	42.1	49.9	59.4	4.0	2.2	4.2	5.5	7.9
RetinaNet	ResNet-101	38.7	22.1	42.9	50.5	58.4	3.9	2.3	4.0	5.6	7.6
Concept RetinaNet	ResNet-101	39.6	24.4	43.5	51.8	60.2	4.1	2.5	4.4	5.7	8.3
DynamicHead	ResNet-50	44.1	27.2	47.8	57.2	61.9	5.6	3.1	5.8	8.1	10.0
Concept DynamicHead	ResNet-50	49.2	31.7	52.8	64.6	69.6	8.9	4.7	9.0	12.6	16.4
DynamicHead	ResNet-101	44.1	26.7	47.7	57.2	61.6	5.7	2.9	6.1	8.2	10.1
Concept DynamicHead	ResNet-101	49.3	33.4	52.6	64.4	69.7	9.2	4.8	9.2	12.8	16.9
DynamicHead	Swin-Tiny	49.9	32.8	53.7	63.9	68.4	6.7	3.6	7.0	9.7	11.7
Concept DynamicHead	Swin-Tiny	53.8	38.1	57.3	68.4	74.2	10.1	5.5	10.3	14.1	18.3

Table E.7: Object detection results obtained with the new sampling strategy and with $d = 1$. *AP (%)* refers to the object detection *mean Average Precision*, while *AP50 (%)* refers to the object detection *mean Average Precision* with *IoU* ≥ 0.5. The models conditioned with the concepts are highlighted with the dove gray color.

Depth Value	AP (%)	AP50 (%)	N. of Concepts
0	39.4	60.9	80
1	38.4	59.4	954
2	38.1	58.5	2586
3	37.7	58.0	5054
4	37.9	58.2	7274

Table E.8: Object detection results using the new sampling strategy varying the concept depth values used for generating the WordNet concepts. The values are obtained using the proposed concept-conditioned RetinaNet model with ResNet-50.

in most of all architecture and backbones combinations, with the only exception of Concept RetinaNet with ResNet-101 on the Focused Visual Genome dataset. Again, these results can be linked to the model selection absent on the Visual Genome.

Both datasets achieve the best AP results by deploying DynamicHead with the Swin-Tiny backbone. On Focused COCO, the larger AP improvement (2%) is given by Concept DynamicHead (51.0%) over DynamicHead (49.0%), both with ResNet-50. While, on Visual Genome, the larger AP improvement (2.3%) is given by Concept DynamicHead (13.0%) over DynamicHead (10.7%), both with ResNet-101. However, in this case, the improvements achieved on the Focused Visual Genome dataset by the conditioned models are higher than those achieved in the Focused COCO. Note that only on Focused Visual Genome, Concept RetinaNet performs slightly worse than the standard version, which could be explained by the non-exhaustive search of hyper-parameters performed during model selection. Appendix E.2 reports more details about the model selection.

In conclusion, even using this new sampling strategy, conditioning the object detection with the user's intent improves the detection performance of an object detector.

Meta Architecture	Backbone	Focused COCO (AP%)					Focused Visual Genome (AP%)				
		AP(%)				AP50 (%)	AP(%)				AP50 (%)
		All	Small	Medium	Large	All	All	Small	Medium	Large	All
RetinaNet	ResNet-50	40.6	25.9	43.6	52.1	63.3	6.9	4.2	6.8	9.1	13.6
Concept RetinaNet	ResNet-50	41.3	28.0	44.4	53.0	64.2	6.8	3.6	6.8	9.1	13.4
RetinaNet	ResNet-101	43.1	27.3	46.9	55.1	65.8	7.4	4.3	7.4	9.8	14.4
Concept RetinaNet	ResNet-101	43.2	29.1	46.6	55.6	65.9	7.2	4.5	7.3	9.7	14.3
DynamicHead	ResNet-50	49.0	31.7	51.9	62.7	69.2	10.7	5.6	10.5	14.2	19.0
Concept DynamicHead	ResNet-50	51.0	33.7	53.9	65.9	71.8	13.0	7.1	12.3	18.1	23.7
DynamicHead	ResNet-101	49.2	32.0	52.6	62.6	69.3	10.9	5.4	10.6	14.7	19.4
Concept DynamicHead	ResNet-101	50.9	34.7	53.4	65.5	71.7	13.2	7.1	12.6	17.9	23.8
DynamicHead	Swin-Tiny	54.3	37.7	57.2	67.4	74.5	12.8	6.6	12.1	17.1	22.0
Concept DynamicHead	Swin-Tiny	55.5	39.9	58.3	69.4	76.0	14.7	7.9	14.2	20.0	26.1

Table E.9: Object detection results obtained with the new sampling strategy and with $d = 1$. All the object detectors are coupled with the *Post-processing Selection* component. *AP (%)* refers to the object detection *mean Average Precision*, while *AP50 (%)* refers to the object detection *mean Average Precision* with *IoU* ≥ 0.5. The models conditioned with the concepts are highlighted with the dove gray color.

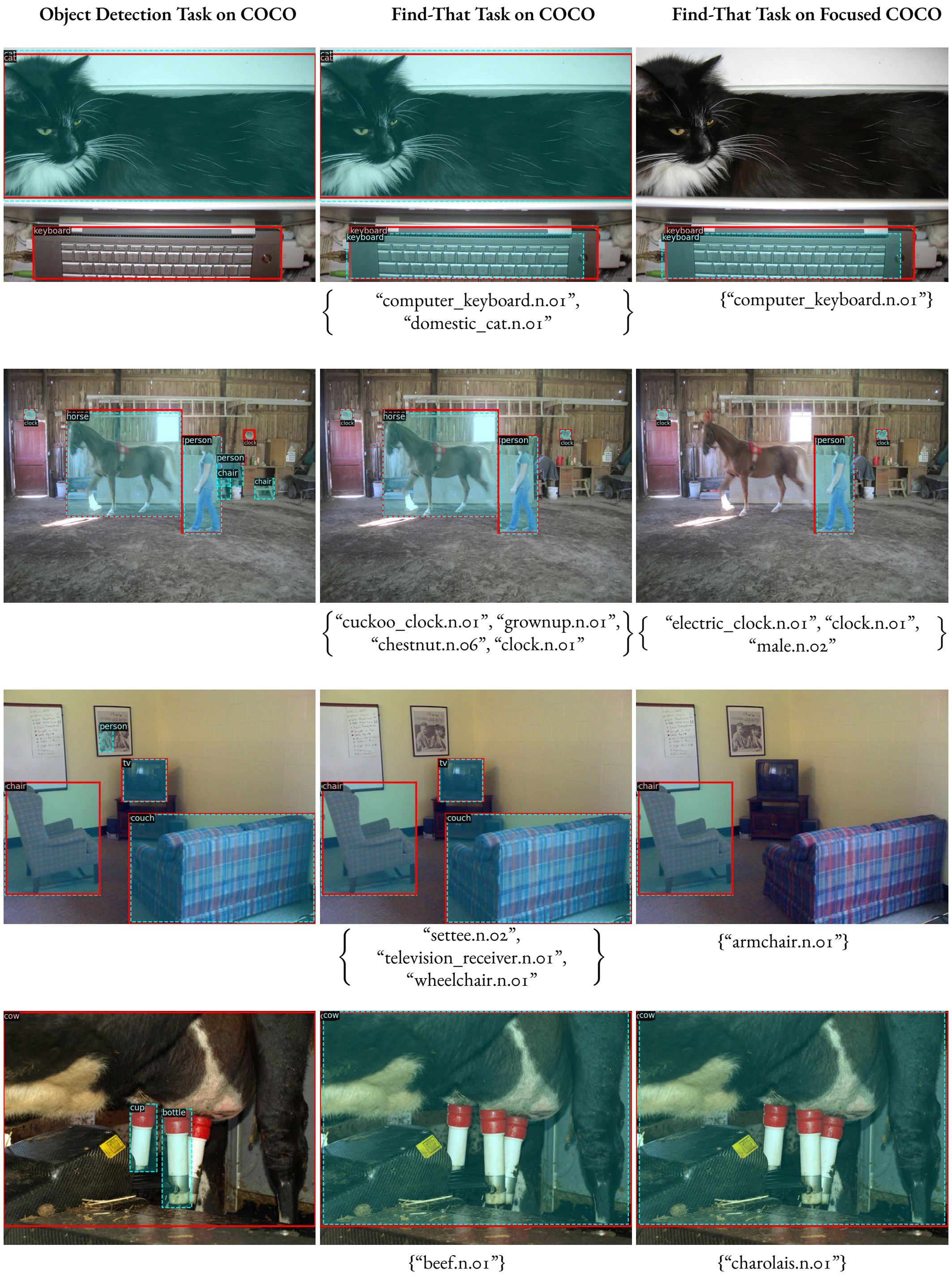

Object Detection Task on COCO
Find-That Task on COCO
Find-That Task on Focused COCO
cat
keyboard
{“computer_keyboard.n.01”, “domestic_cat.n.01”}
{“computer_keyboard.n.01”}
horse
person
chair
clock
{“cuckoo_clock.n.01”, “grownup.n.01”, “chestnut.n.06”, “clock.n.01”}
{“electric_clock.n.01”, “clock.n.01”, “male.n.02”}
chair
tv
couch
{“settee.n.02”, “television_receiver.n.01”, “wheelchair.n.01”}
{“armchair.n.01”}
cow
cup
bottle
{“beef.n.01”}
{“charolais.n.01”}

Figure E.3: Examples of predictions obtained with the proposed model. It is delimited with red lines the ground truth bounding boxes and with light blue dashed lines the model’s predictions.

Appendix F
Publications

This document presents the results of the research activities performed throughout the duration of the doctorate program, which has led to a series of publications at international conferences summarized below.

F.1 International Conferences

- **D. Rigoni**, D. Elliott, and S. Frank, "Cleaner Categories Improve Object Detection and Visual-Textual Grounding", in *Image Analysis: 23rd Scandinavian Conference, SCIA 2023, Sirkka, Finland, April 18–21, 2023, Proceedings, Part I (pp. 412-442).* Cham: Springer Nature Switzerland.

- **D. Rigoni**, L. Serafini, and A. Sperduti, "A better loss for visual-textual grounding", in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 49–57.

F.2 Under Peer Review

- **D. Rigoni**, L. Serafini, and A. Sperduti, "Object Search by a Concept-Conditioned Object Detector", *Under Peer Review*.

- **D. Rigoni**, L. Parolari, L. Serafini, A. Sperduti, and L. Ballan, "Weakly-Supervised Visual-Textual Grounding with Semantic Prior Refinement", *Under Peer Review*.