Master-Arbeit am Institut für Computerlinguistik der
Universität Heidelberg

Employing the Scene Graph for Phrase Grounding

Analysis, Enhancements and ConceptNet Extensions

Supervisorin:
Prof. Dr. Anette Frank

Verfasserin:
Julia Suter
Matrikelnummer 3348630

21. September 2020

Abstract

The multimodal task of phrase grounding links phrases in a caption to the regions in the image they refer to and is essential for many applications that require a connection between the visual and textual modality. In this thesis, we adapt the phrase grounding system introduced by Parcalabescu and Frank [2020], which employs the scene graph as a representation for images and aligns phrases to scene graph objects using the cosine similarity derived from the word embeddings of phrase and object label. We perform a comprehensive analysis of the scene graph representation and phrase grounding system in order to discover their capabilities and limitations and identify ways to improve them. We enhance the scene graph representation by constructing plural objects, removing superfluous objects and adding two attributes, object color and object location in foreground or background. We extend the phrase grounding system with methods that filter out candidates based on color, number of represented instances, and the age, gender and position of people. Collectively, these enhancements result in a 6% increase in phrase grounding accuracy. We also suggest methods for integrating external world knowledge from ConceptNet and prototype an approach based on extracting support concepts for caption phrases and scene graph labels, which is however impeded by limitations inherent in ConceptNet and thus does not increase performance in its current form. Finally, we develop two alternative similarity measures derived from caption-specific ConceptNet subgraphs that when combined are comparable to the word embeddings similarity measure, indicating that ConceptNet does contain relevant information that can be harnessed through appropriate strategies. Importantly, we also observe throughout our work that the conventional evaluation metric for phrase grounding is biased toward recall and does not always provide adequate assessments.

Zusammenfassung

Phrase Grounding bezeichnet die Verknüpfung einer Phrase aus einer Bildunterschrift mit jenem Ausschnitt des Bildes, welchen sie bezeichnet. Phrase Grounding ist essenziell für viele Anwendungen, die eine Verbindung zwischen visueller und textueller Modalität benötigen. In dieser Thesis erweitern wir das System für Phrase Grounding von Parcalabescu and Frank [2020], welches den Scene Graph als Repräsentation des Bildes nutzt und Phrasen mit Objekten aus dem Scene Graph verbindet, basierend auf der Cosine Similarity der Word Embeddings von Phrase und Bezeichnung des Scene Graph Objektes. Wir führen eine umfassende Analyse des Scene Graph und des Systems für Phrase Grounding durch, um deren Potential und Grenzen auszuloten und Verbesserungsmöglichkeiten zu finden. Wir reichern den Scene Graph mit zusätzlicher Information an, indem wir Plural-Objekte konstruieren, überflüssige Objekte entfernen und Attribute hinzufügen, die Farbe und Position im Vorder- oder Hintergrund repräsentieren. Wir erweitern das System für Phrase Grounding mit Methoden, um Kandidaten anhand von Farbe, Anzahl der repräsentierten Objekte sowie im Falle von Menschen Alter, Geschlecht und Position herauszufiltern. Gemeinsam erwirken diese Verbesserungen eine Erhöhung der Accuracy um 6%. Wir stellen auch Methoden vor, die externes Weltwissen aus ConceptNet integrieren und entwickeln einen Ansatz, der Hilfskonzepte für Phrasen und Bezeichnungen von Scene Graph Objekten extrahiert. Dies wird allerdings erschwert durch die Grenzen von ConceptNet, weshalb damit momentan noch keine Verbesserung erzielt wird. Schließlich entwickeln wir zwei alternative Ähnlichkeitsmaße, die aus den für jede Bildunterschrift einzeln generierten ConceptNet-Subgraphen extrahiert wird. Beide kombiniert erzielen vergleichbare Resultate mit dem Ähnlichkeitsmaß für Word Embeddings, was veranschaulicht, dass ConceptNet relevante Informationen enthält, die durch geeignete Strategien erschlossen werden können. Darüber hinaus beobachten wir in verschiedenen Aspekten unserer Arbeit, dass die konventionelle Evaluationsmetrik für Phrase Grounding verzerrt ist und nicht immer eine angemessene Beurteilungen bietet.

Acknowledgements

I would like to thank all the people who helped and supported me while I was working on my Master thesis.
First and foremost, I would like to thank Prof. Dr. Anette Frank who supervised my work. She helped me develop interesting research questions and encouraged me not to lose sight of the overall picture while working on the details. I am grateful for her advice, feedback and patience.
I would also like to thank Prof. Dr. Michael Strube, my second supervisor. Our joint work on the entity graph showed me the power of graph theory and inspired some of the ideas presented in this thesis.
Special thanks go to Letitia Parcalabescu who introduced me to the topic of phrase grounding. Her system provided an invaluable basis for my thesis project. She was also always quick to help me out when I had questions.
I would like to thank my partner Jonas Hartmann for supporting and encouraging me throughout the research and writing process. I am especially grateful for the interesting discussions and the proof-reading of the entire thesis.
My friends have been a great moral support during the Master thesis. Special thanks go to Elisa Gallo who has been writing her own thesis at the same time and always motivated me to keep going.
Last but not least, I would like to thank my family, especially my mother, for supporting and encouraging me throughout my studies.

Table of Contents

1 Introduction

Information can be conveyed in a variety of ways, for instance through a picture, a news article or an audio recording. Image, text and speech represent different modalities, where a mode is defined as a resource for making meaning that is shaped through culture and society [Kress, 2010, p.79]. A medium that includes more than one mode expresses its information through multimodality, for instance a video clip. Similarly, if a task requires information from several modes in order to be solved (e.g. visual and textual input), it is considered a multimodal task. Different modalities represent different aspects of a medium and thus complement each other, so the essence of any multimodal task is to meaningfully link the information conveyed by the different modalities.

Figure 1.1 gives an example for how the modalities vision and language can represent different information. The image shows a little pedestrian road between traditional Japanese buildings and shops. There are people walking down the road or standing in front of shops, some are holding an umbrella. The caption reads *Tourists walking down a road with shops on both sides after visiting the Kiyomizudera temple, August 2013*. There is some information represented by both modalities, for example that there are people walking down a road with shops on either side. However, both modalities contain exclusive information as well: from the image we learn the number of people present, their appearance and exact positions. This information is not provided by the caption, although it specifies that the people are tourists. The caption also explains where the tourists are coming from, something that cannot be inferred from the image alone. It also gives a date, which is impossible to pinpoint by looking solely at the picture. This example shows that two modalities representing the same scene do not necessarily have the same focus and do not describe the same information. There is information that can more easily be described by the visual modality (positions of objects and people) and information that is better described by the textual modality (scene context, date). For better interpretation, either modality benefits from additional knowledge provided by knowledge resources.

The style of the buildings as well as the Japanese characters on the shop banners suggest that it is a place in Japan. The caption, however, does not explicitly state

Figure 1.1: *Tourists walking down a road with shops on both sides after visiting the Kiyomizudera temple, August 2013.*

that the picture was taken in Japan. In a further step, one could extend the information given by the caption with world knowledge from knowledge resources. If the location of the Kiyomizudera temple is looked up, we can not only confirm that the picture was indeed taken in Japan, but specifically in Kyoto.

The interpretation of images not only requires factual knowledge (such as locations of famous temples) but also common-sense and cultural knowledge. An example for common-sense knowledge is that a person with sunglasses and a camera at a tourist hotspot is most likely a tourist. An example for cultural knowledge is that in Asian countries it is common for people to carry an umbrella in sunny weather in order to shield themselves from UV rays. Thus, the fact that there are several opened umbrellas does not necessarily mean it is raining.

Phrase grounding constitutes a perfect example of a multimodal task that connects the modalities vision and language. *Phrase grounding* or *phrase localization* describes the task of aligning phrases in the caption with the corresponding region of the image. A *phrase* is defined as a segment of a caption (usually a noun phrase) that describes one specific object or area in the image. Regions are usually marked by *bounding boxes* enclosing the object in question. Figure 1.2 gives an example for phrase grounding: the phrases *a fluffy golden puppy*, *a yellow toy* and *its mouth* have to be aligned to their corresponding (same color) bounding boxes in the image.

Figure 1.2: Phrase grounding example.

Grounding the phrase with its image region provides an important link between the visual and textual modality and enables knowledge and context transfer between them. Phrase grounding provides an essential first step for many other multimodal understanding tasks that depend on a connection between image and text, including visual question answering, sentence-to-image alignment, visual common-sense reasoning and visual information retrieval. Thus, it has become an established research task [Kazemzadeh et al., 2014; Plummer et al., 2015; Krishna et al., 2017].

The greatest challenge of phrase grounding is presented by the phrases in free text form. While object detectors assign a category from a closed vocabulary (e.g. *dog*), the phrases describe the object with any vocabulary the annotator finds suitable (e.g. *a fluffy golden puppy*). Focus, degree of specificity and attention to detail are at the discretion of the annotator, meaning that multiple captions for the same image can describe different aspects in different ways.

The majority of previous work on phrase grounding pursues strongly or weakly supervised approaches, requiring a large number of annotated phrase-region pairs. To the best of our knowledge, there exist only two unsupervised approaches for phrase grounding [Wang and Specia, 2019; Parcalabescu and Frank, 2020]. As they do not require a large amount of training data, they remain domain-independent and transparent. Parcalabescu and Frank [2020] employ the scene graph to represent the image, and align phrases to scene graph objects using similarity rankings derived from word embeddings. They also add methods for contextualizing the visual and textual modalities.

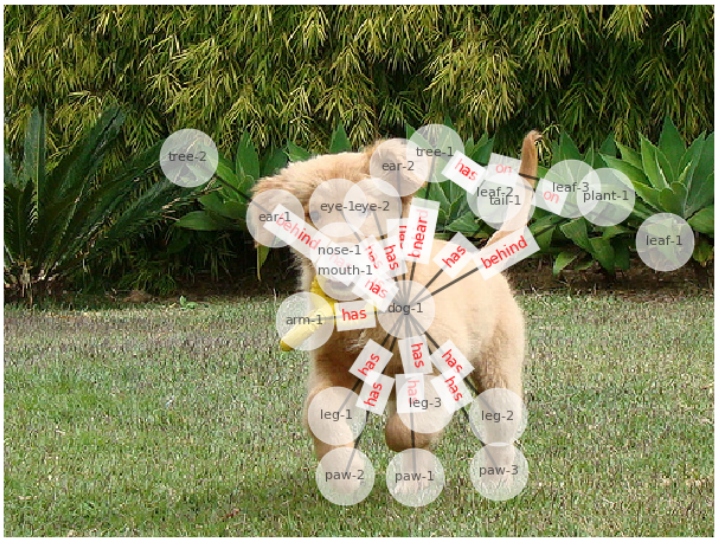


Figure 1.3: Scene graph example.

The *scene graph* [Johnson et al., 2015] represents an image as a graph with detected objects as nodes and their relations as edges. It puts the objects of a scene in context with each other. Figure 1.3 shows the scene graph for the example picture in Figure 1.2 overlaid on the image. In the center, there is the object node representing *dog*, which has many outgoing edges to body parts, for instance *dog-1 has leg-1* or *dog-1 has eye-2*. The vegetation in the background is identified as trees, plants and leaves. The objects in the foreground and background are connected by edges such as *dog-1 near tree-1* and *plant-1 behind dog-1*. Figure 3.3 provides another example of a scene graph.

In this thesis, we will report our work to enhance the scene graph representation and adapt the unsupervised system introduced by Parcalabescu and Frank [2020] in order to improve phrase grounding performance. We work with the base system without contextualization of image and language representation, which enables us to better study the effects of the invididual enhancements. We pursue three key research goals. First, we seek to determine the quality of the automatically generated scene graphs and to discover the challenges an unsupervised phrase grounding system using the scene graph has to master. Second, we aim to find ways to enhance both the scene graph representation and the phrase grounding system in order to achieve higher quality scene graphs and improved performance. And third, we endeavor to develop methods for including external world knowledge through ConceptNet and study its effects on the phrase grounding system.

For the first goal, we conduct an in-depth analysis of the scene graph representation. We examine how well it represents the images in the Flickr30k Entities dataset and what information is not yet captured in it. We suggest methods for retrieving missing information and generally improving the quality of the scene graph. Finally, we analyze the phrase grounding performance of the system described in Parcalabescu and Frank [2020] and discuss its shortcomings. We also take a critical look at the evaluation metric.

Regarding the second goal, we use the insights gained from the analysis in order to improve the scene graph representation and the phrase grounding system. We enhance the representation by adding plural objects and attributes and by removing superfluous objects in order to achieve a richer and cleaner representation. We also extend the phrase grounding system by adding methods for filtering object candidates based on information extracted from the phrase and scene graph objects, for instance color or whether they represent one or several objects. We evaluate the enhancements on the phrase grounding task and compare the different approaches.

Finally concerning the third goal, we exploit ConceptNet as a resource for external knowledge, both by incorporating knowledge into the phrase grounding system and by directly grounding phrases through similarity measures based on graph metrics derived from ConceptNet. We generate ConceptNet subgraphs for each caption/image pair in order to facilitate processing and to extract more accurate concepts and graph metrics. We evaluate the ConceptNet extensions and discuss our findings.

The thesis is structured as follows: Chapter 2 provides an overview of previous work in the field of phrase grounding, with a particular focus on the approach suggested by Parcalabescu and Frank [2020], on which this work is based. In Chapter 3, we conduct a thorough analysis of the scene graph representation, discuss what information is missing and how it could be retrieved, and evaluate the performance of the phrase grounding system we attempt to enhance in further steps. In Chapter 4, we describe how to improve the scene graph representation by adding plural objects and attributes and removing superfluous objects. We also introduce three extensions to the phrase grounding system to filter out unsuitable candidates. Chapter 5 describes the ConceptNet extensions that add external knowledge to the system, as well as the ConceptNet subgraph similarity measures used for phrase grounding. In Chapter 6 we revisit these various approaches and results and further discuss our findings. Finally, we end on some general conclusions in Chapter 7.

2 Previous Work

Phrase grounding or phrase localization is an essential first step for many multimodal tasks (e.g. visual question answering) as it provides a link between the visual and textual modality. Thus, it has become an established research task and large annotated datasets have been created for training and testing phrase grouding systems [Plummer et al., 2015; Russakovsky et al., 2015; Chen et al., 2015].

An essential prerequisite for phrase grounding is accurate object recognition: the task of detecting objects in an image, creating a bounding box surrounding them, and selecting a label from a fixed set of object labels. The performance of object detectors is rapidly improving, owing to large datasets and the development of deep convolutional neural network systems [Sermanet et al., 2013; Girshick et al., 2014; Girshick, 2015; He et al., 2015; Liu et al., 2016; Redmon et al., 2016; Redmon and Farhadi, 2017]. However, while high-quality output of the object detector is crucial for phrase grounding, the real challenge lies in mapping the open-vocabulary phrases to the detected objects.

Phrase grounding approaches are generally classified as *strongly supervised*, *weakly supervised* or *unsupervised*. Strongly supervised models require annotated pairs of phrases and image regions [Guadarrama et al., 2014; Hu et al., 2016; Rohrbach et al., 2016; Wang et al., 2016; Chen et al., 2017a,b; Hinami and Satoh, 2017; Zhang et al., 2017; Plummer et al., 2017, 2018; Wang et al., 2018; Dogan et al., 2019; Lu et al., 2019; Li et al., 2019; Sadhu et al., 2019]. Weakly supervised approaches use pairs of phrases and entire images; the ground truth location of the object in the image is not given [Karpathy et al., 2014; Xiao et al., 2017; Chen et al., 2018; Yeh et al., 2018; Zhao et al., 2018]. Unsupervised models require no paired training examples at all [Wang and Specia, 2019; Parcalabescu and Frank, 2020].

2.1 Supervised Phrase Grounding

There are various methods for strongly supervised phrase grounding: some embed both image region and phrase into vectors and project them onto a common space in order to measure similarity between text and image [Plummer et al., 2017; Wang

et al., 2016, 2018], others attempt to reconstruct the phrase from the image region [Rohrbach et al., 2016]. Hinami and Satoh [2017] tackle the issue of open-vocabulary by training an object detector on phrases rather than fixed classes. Dogan et al. [2019] do not ground phrases independently from each other but as a sequential and contextual process. The most recent systems process and combine text and image in multiple transformer layers [Lu et al., 2019; Li et al., 2019].

In weakly supervised models the ground truth localization is not provided. Thus, one needs to integrate methods for localization such as (knowledge-aided) external region proposals [Chen et al., 2018; Zhao et al., 2018] or region proposals derived from caption-to-image retrieval [Datta et al., 2019]. Generic object category detectors [Yeh et al., 2018] and spatial attention masks [Xiao et al., 2017] have also been used to find candidate proposals. As for learning how to link a phrase to the correct regional proposal, there exist several approaches: Xiao et al. [2017] analyze the parse tree of the caption in order to extract information about complementarity and compositionality of the candidate regions. Zhao et al. [2018] use region proposals as anchors for a continuous search over the spatial feature map. Yeh et al. [2018] link words to image concepts using co-occurrence statistics. Chen et al. [2018] compute language and visual consistency by comparing the original and reconstructed caption and comparing the candidate region with the reconstructed proposal.

2.2 Unsupervised Phrase Grounding

To the best of our knowledge, there exist only two entirely unsupervised methods for phrase grounding [Wang and Specia, 2019; Parcalabescu and Frank, 2020]. According to Wang and Specia [2019], unsupervised models are more similar to how human localize objects in images: not by learning pairs but by combining several tasks and resources (e.g. object detection, semantic knowledge of phrases, knowledge bases). Furthermore, unsupervised models have a number of key advantages outside of accuracy. Namely, they are domain-independent and do not rely on a large number of training samples. They are comparably simple and interpretable, which makes the prediction process transparent. And finally, they are suitable as a strong baseline for supervised approaches.

Wang and Specia [2019] combine the results of four different object detectors in order to retrieve a large number of labeled candidate bounding boxes. They also include a color detector. Then, each bounding box label and phrase from the caption is embedded as a word vector. For grounding a phrase, they compute the cosine similarity between the given phrase and all candidate bounding box labels and se-

lect the label with the highest similarity. Parcalabescu and Frank [2020] call this a *bag-of-objects approach*. The system by Wang and Specia [2019] outperforms some of the weakly supervised approaches and is competitive against the strongly supervised approaches they compare with. More recent work with strongly supervised transformer models, however, clearly outperform their system [Li et al., 2019].

Parcalabescu and Frank [2020] pursue a similar approach for unsupervised phrase grounding, but incorporate methods for overcoming some of the shortcomings of the approach reported by Wang and Specia [2019]. First of all, they employ a single object detector, rather than a collection of four which ensures a fair comparison to other systems with a single detector. In Wang and Specia [2019], accuracy drops by 6% (from 50.49% to 44.69%) when only the best performing detector is used rather than the best combination of all four. Furthermore, by employing a single detector with a more fine-grained label set (1600 labels), Parcalabescu and Frank [2020] established a harder but more realistic proposal upper bound of 87.9%, which is 37% higher than the upper bound reported in Wang and Specia [2019]. Finally, instead of using a bag-of-words model, Parcalabescu and Frank [2020] use a structured visual representation of the detected objects in the form of a scene graph.

The scene graph was first introduced by Johnson et al. [2015] for image retrieval and has since been used as an aid in tasks that process both natural language and visual input [Johnson et al., 2015; Teney et al., 2017]. The scene graph represents real-word images in a graph structure, providing a semantic overview of objects and their relationships, which helps in contextualizing the visual modality. Figure 3.3 shows an example of a scene graph. Chapter 3.2 will explain how the scene graph is constructed in more detail.

Parcalabescu and Frank [2020] do not only contextualize the visual modality using the scene graph, they also contextualize the labels of the detected objects in the linguistic modality. The motivation behind this is that the cosine similarity of the non-contextualized words often fails to reflect the actual similarity between two words. The word *groom*, for instance, is closer to *woman* than to *man*, so it would be aligned incorrectly. They enhance the representation of each label by aggregating it with neighboring concept meanings. The neighboring concepts are retrieved from WordNet and Open Images (OI). The contextualized meaning is computed by taking the mean of the embeddings of the original label and its neighboring concepts, with the idea that the vector is shifted into the right direction. Finally, they introduce an alternative similarity measure to cosine similarity by identifying the shortest path on the WordNet graph structure. They test their model on the Flickr30k Entities dataset and outperform all weakly supervised models and several supervised models (excluding transformer models) with an accuracy of 57.08%.

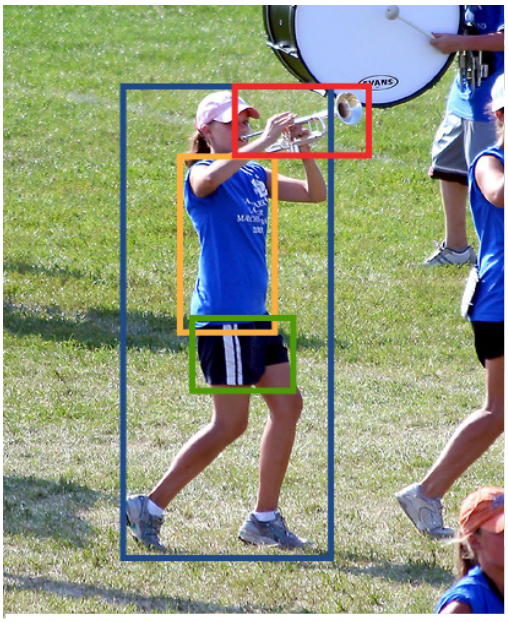
3 Analysis

As one of the goals of our work is to enhance the scene graph representation, we begin by analyzing the original scene graph in order to identify what can be improved. Since we test the phrase grounding performance on the Flickr30k Entities dataset, it is relevant to also examine the nature of the Flickr30k images and captions. Thus, in this chapter we investigate step by step the Flickr30k dataset, the scene graph representation of the Flickr30k images, both in general and in regards to the phrase grounding task, and finally the phrase grounding performance based on these Flickr30k image scene graph representations, following the approach by Parcalabescu and Frank [2020].

3.1 Flickr30k Entities Dataset

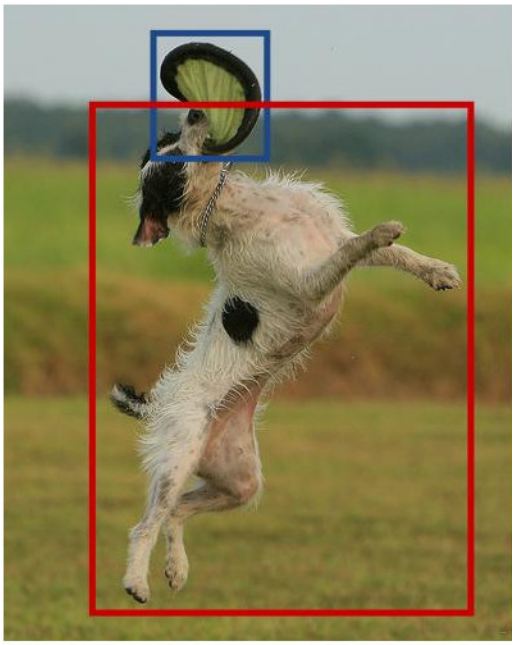
The Flickr30k dataset [Young et al., 2014] contains 31783 images with 5 English captions each (158915 captions in total) and has been used for training and evaluating caption generation and retrieval systems. Figures 3.1 and 3.2 show examples of Flickr30k images with their 5 captions. The images are of different format, size and quality; some images are blurry or poorly lit. The pictures are often snapshots and the depicted people were not aware that an image was being taken, so they do not necessarily face the camera and are often caught in mid-motion. Since the captions are manually created by different annotators, the captions can show different focus, interpretations and levels of specificity. For the image in Figure 3.1 some captions describe the clothing in detail while others do not mention it at all. The third caption assumes that the picture was taken in context of a sports game, although there is no clear evidence for this. Some captions contain spelling errors (e.g. *trumped* instead of *trumpet*).

In the more refined Flickr30k Entities dataset [Plummer et al., 2015], the phrases within the captions, also called entities, are identified and labeled with a unique entity ID. Phrases are chunks of the captions (usually noun phrases) that describe one specific object in the image, for instance *a young girl* or *a frisbee*. Coreferent phrases across captions for the same image share an ID, for instance the dog (marked



A young girl wearing a blue shirt marching in a band playing a trumpet.
Girl wearing blue shirt and black shorts plays trumped outside.
A teenager plays her trumpet on the field at a game.
A girl playing trumpet in a marching band.
Girl playing the trumpet in a marching band.

Figure 3.1: Flickr30k Entities example.



A white dog is leaping in the air with a green object in its mouth .
A terrier mix catches a Frisbee in the air .
A white and black dog catching a Frisbee .
A dog catches a Frisbee in midair .
A dog catching a Frisbee .

Figure 3.2: Flickr30k Entities example.

in red) in Figure 3.2. Captions 4 and 5 refer to the dog simply as *a dog*, captions 1 and 3 call it *a white dog* and *a white and black dog*, while caption 3 specifies that the dog is *a terrier mix*.

Each entity is assigned a manually annotated bounding box that surrounds the area of the image in which the entity is shown, 275k bounding boxes total. Thus, the Flickr30k Entities dataset provides ground truth information for matching phrases in the captions with regions in the image and it has become a standard benchmark for training and testing of phrase grounding systems.

On average, a caption contains 3.3 phrases, represented with as many bounding boxes. Within a set of 5 captions for one image, there are on average 16.6 phrases, yet only 7.7 unique entity IDs, indicating that the different captions often refer to the same objects in the image. Because of this overlap, there are on average half as many bounding boxes as there are phrases. On average, 77% of all phrases in a caption set are mentioned in multiple captions.

The Flickr30k Entities dataset divides the phrases into 7 categories: *people, scene, clothing, body parts, animals, instruments* and *other*. Table 3.1 shows the distribution of these categories across the dataset. We also report the image coverage, i.e.

the percentage of images that include at least one phrase from a given category. The category *people* is by far the most frequent: almost every image contains at least one person and the category makes up 34% of all phrases. This is an important insight since images of people have different characteristics than images of landscapes or inanimate objects. For images containing people, the caption usually describes what these people look like and what they are doing. Furthermore, the grammatical subject of Flickr30k captions is usually a person, for exmaple *Two men in green shirts are standing in a yard.* Knowing what information and structure to expect from the caption can be helpful when building a phrase grounding system. If there is no person in the image, there is usually an animal depicted. Only 0.04% of all images contain neither.

The *other* category is most diverse and includes objects such as *table* or *microphone* as well as more abstract concepts such as *music* or *a good time.* In 24% of all cases, no bounding box was assigned to the phrase, either because the object is not visible in the picture or it is too abstract. The same can be observed for the *scene* category: 41% of all *scene* phrases do not have a bounding box, usually because the scene is represented by the entire image. The term *scene* is not clearly defined and can include specific locations or objects such as *building*, *street*, *beach*, *garden*, *fence* or even *bus* (bounding box exists) or more abstract concepts such as *store*, *parade*, *party*, *concert* or *mid-air* (no bounding box).

The category *clothing* is relatively frequent as well: for 70% of all images some sort of clothing is described. The *body part* category on the other hand is not as frequent, although we know that nearly every image contains people – and thus body parts. This shows that what is depicted in the image is not necessarily represented in the caption, especially if it is something one can infer with common sense. A person – by definition – has a head, two eyes, a nose, two arms and legs so there is no need to explicitly mention them unless they are relevant for the action, for example *two people shaking hands.*

Animals, vehicles and instruments occur only in every tenth image and combined make up only 6% of all phrases. 67% of the animals in the dataset are dogs so they are highly over-represented.

As for the nature of the phrases, they are very diverse and of different length. When considering only the last (unlemmatized) word in the phrase, which usually constitutes the head of the noun phrase, there are 11k different phrase heads in a total of 529k phrases. The most frequent phrase heads are: *man, woman, people, shirt, girl, men, boy, dog* and *street.*

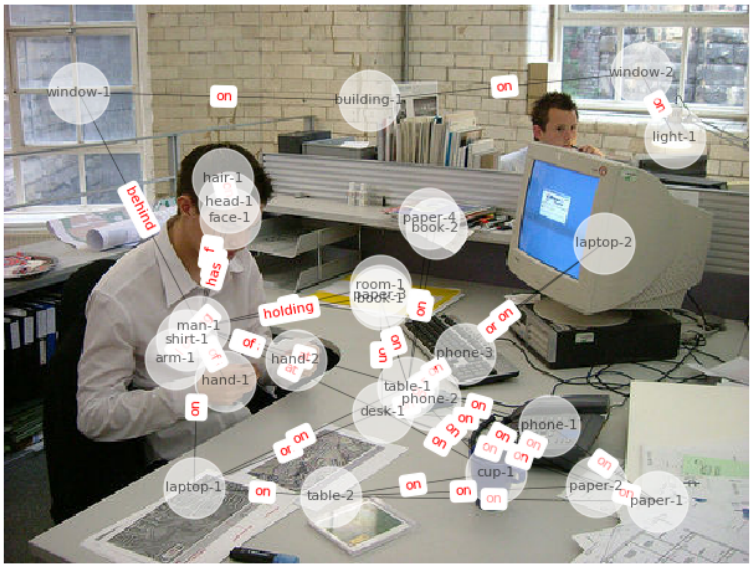


Figure 3.3: Scene graph overlaid on image.

3.2 Scene Graph Representation

The scene graph is a data structure that represents a scene in an image [Johnson et al., 2015] by encoding object instances and their relationships. The nodes in the scene graph represent the detected objects in an image. Each object node contains information about the location in the image (i.e. bounding box) and a label describing the object. The labeled edges in the scene graph represent the (visual) relation between the objects. As in Parcalabescu and Frank [2020], we employ the readily available scene graph generator of Zellers et al. [2018], which is trained on Visual Genome [Krishna et al., 2017] with the most frequent 150 object labels and 50 relation labels [Xu et al., 2017]. While Parcalabescu and Frank [2020] generate the scene graph by using the 50 most confident relationship triples yielded by the generator, we use all of them. On average, there are 1152 triples for one image. Figure 3.3 shows an example for a scene graph, overlaid on the image it originated from.

3.2.1 Categories and Labels

The 150 object labels include 14 bodyparts, 13 clothing items, 9 terms for people, 10 animals, 10 vehicles and 94 others. Table 3.1 shows the distribution and image coverage of the categories in comparison to the ground truth Flickr30k annotations. When analyzing the differences, one should bear in mind that they do not represent

the same modality: the Flickr30k ground truth represents captions while the scene graph represents images. Not everything shown in a picture is also mentioned in a caption, which (partially) explains why there are 10 times as many scene graph nodes (2021k) as there are ground truth boxes (243k). Nevertheless, the table allows for some interesting observations. The category *other* is clearly the most frequent: 94 labels classify as *other* and make up roughly half of all scene graph nodes. The next most frequent category is *body parts* (20%). *Body parts* are found in nearly every picture but they are rarely mentioned in the captions. Similarly, *clothes* are represented in almost every scene graph but not in as many captions.

While the relative distribution suggests that people nodes are more than twice as frequent in the Flickr30k captions as in the scene graph, the absolute numbers reveal that the scene graphs include almost 5 times as many people nodes as captured by the ground truth. This can be explained a) by people in the background that are included in the scene graph but not mentioned in the caption, b) plurals (*e.g. three children*) and other collective terms (e.g. *crowd* or *group*), which yield one phrase in the caption but several nodes in the scene graph, and c) multiple objects for the same person (discussed in detail in Chapter 3.2.3). Similar reasons may apply to *animals* and *vehicles*, for which the image coverage is higher in the scene graph than in the ground truth.

The 150 scene graph labels do not include any *scenes*, as *scenes* are usually not represented by a single detectable object. *Instruments* are also missing in the scene graph label set, possibly because they are too specific and thus rare.

The 10 most frequent scene graph labels are *shirt, man, tree, person, head, hand, leg, hair, woman* and *arm* (5 body parts, 3 people, 1 clothing, 2 others). Combined these 10 labels make up one third of all scene graph nodes. The top 20 labels cover 52%. This indicates that the distribution across the labels is highly skewed. On the tail end, there are 30 labels that make up less than 0.1% of all nodes, including *windshield, fork, skier* and *zebra*. There are also some redundancies among the 150 labels, for instance *plane* and *airplane*, *hat* and *cap*, *child* and *kid*, or *woman* and *lady*. This does not seem to pose an immediate problem for image representation itself but can reduce performance and efficiency of more complex tasks.

In order to investigate how well Flickr30k phrases are covered by the scene graph, we count for how many Flickr30k bounding boxes there exists a suitable scene graph object of the same category with a similar bounding box. A similar bounding box is defined as an intersection over union (IoU) score $>= 0.5$. We call this score *Flickr30k coverage*. Note that this measure is merely an approximation, as it only considers categories and not labels. Furthermore, an overlap of two bounding boxes

	Ground Truth		Scene Graph		Flickr30k Coverage		
Category	Distr.	Img cov.	Distr.	Img cov.	Cat.$_{w/}$	Cat.$_{w/o\ plurals}$	No cat.$_{w/o}$
people	34.4	94.1	13.2	95.3	70.7	88.4	96.4
other	26.2	91.8	47.4	100.0	58.8	59.0	79.2
scene	15.6	78.8	-	-	-	-	84.7
clothing	13.5	69.8	16.1	95.4	63.1	72.9	91.5
body parts	3.3	27.5	20.5	96.8	48.8	57.7	68.1
animals	3.2	11.8	1.0	22.1	78.9	83.4	93.2
vehicles	1.9	11.0	1.8	34.1	67.2	71.8	90.8
instruments	0.9	10.9	-	-	-	-	64.6
Total					65.1	72.7	85.9

Table 3.1: Distribution and image coverage for Flickr30k ground truth and scene graph representation; Flickr30k coverage with category constraint (*Cat.*), with and without plural phrases, as well as without category constraint (*No cat.*), without plural phrases.

does not necessarily mean that they represent the same object (see Chapter 3.3.2). We expect the categories *people*, *body parts* and *clothing* to show a high Flickr30k coverage since these categories have an extensive label set and high image coverage. The results, however, turn out to be lower than expected (see Table 3.1, Cat.$_{w/}$). A manual evaluation of 50 samples revealed that 70% of the Flickr30k phrases with no suitable scene graph bounding box are plural phrases, for instance *two girls*. Since the scene graph only represents single objects, there is usually no suitable bounding box for phrases representing multiple objects. When only considering single phrases to compute the Flickr30k coverage, the scores generally increase (see Table 3.1, Cat.$_{w/o\ plurals}$); the overall score increases from 65% to 73%. Discarding plural phrases has the strongest effect on the category *people* (+18%), which ultimately is the category with the highest Flickr30k coverage (88.4%). Animals have a high coverage as well, despite there only being 11 different animal labels. There are several examples that show that animals not listed in the label set (e.g. lion) are still detected and simply labeled with one of the existing animal labels (e.g. bear). This increases recall at the cost of precision. *Clothing* items and *vehicles* are also covered sufficiently with over 70%.

Body parts show a surprisingly low Flickr30k coverage (57.7%), given their frequency in the scene graph and the extensive label set. A look at a subset of failed samples shows that the scene graph does not cover a specific set of body parts that are often mentioned in the captions, for instance *shoulder, hip, lap, belly, chest, waist, butt, body*. These body parts describe diffuse body areas without clearly detectable

boarders, while most scene graph labels represent distinct limbs and parts of the head (e.g. arms, leg, ears). This demonstrates nicely that object detectors identify single clearly separable objects while language often describes images in more vague or generalizing terms. (This can be best observed for scenes: while the object detector identifies several trees, a bench and a pond, the caption may not mention any of these objects and sums it up as a park.)

Finally, we lift the category constraint on the Flickr30k coverage measure and simply compute whether for a given Flickr30k phrase there exists a scene graph bounding box of any category that matches in terms of IoU. The scores for the individual categories with discarded plural phrases can be seen in Table 3.1 in the column *No cat.*$_{w/o}$. The removal of the category constraint results in an overall Flickr30k coverage of 83% when considering all phrases and 86% when discarding plural phrases. This significant increase of 14% can have two possible causes: a) many objects are actually detected but they are labeled wrong and thus fall into the wrong category, and b) the detected bounding box does not actually represent the phrase in question; the IoU measure is simply too forgiving. Inspection of the data provides exemplary evidence for both, supported by the fact that *scenes* and *instruments* are technically not represented by the scene graph but there are supposedly suitable boxes in 84.7% and 64.6% of all cases, respectively (see Table 3.1, No cat.$_{w/o}$). The Flickr30k coverage is an important measure for evaluating a phrase grounding system as it represents the upper bound. If for a given phrase there is no suitable bounding box in the scene graph representation, this phrase can never be grounded correctly. Thus, the upper bound represents the best possible phrase grounding performance with a given image representation. The scene graph representation yields an upper bound of 83.3%.

The Flickr30k ground truth boxes are on average three times larger than the scene graph objects. One explanation is that the phrases can describe several people and objects and the scene graph objects only cover one each. The average size for scene graph objects is also brought down by the large number of very small bounding boxes for body parts and clothes.

The scene graph does not only contain object nodes but also edges representing relations among the objects. On average, there are 64 object nodes and 1152 relation edges per image. The average degree is 18. The 50 relation labels include mostly visual and positional relations (e.g. *on*, *behind*, *in front of*), some passive actions (e.g. *wearing*, *holding*, *sitting on*), and few actual actions (e.g *eating* or *riding*). The top 10 relations are *on*, *has*, *near*, *behind*, *of*, *wearing*, *in*, *holding*, *with* and *under*. The relations *on*, *has* and *near* alone make up 72.8% of all relations. Five of the top 10 overlap with the 10 most frequent relations found in the Flickr30k captions

(*in, on, and, at, of, to, with, wearing, for, in front of*, covering 86% of all caption relations), which suggests that they are a suitable selection. On the other hand, 28 scene graph relations are extremely rare (<0.1% of all relations), 11 of them occur less than 100 times in the entire dataset, for instance *covering*, *made of* and *against*. Some relations are highly overlapping in their use and thus redundant, e.g. *wears* and *wearing* or *lying* at and *laying at*.

The 150 object labels and 50 relation labels were selected because they are the most frequent in the Visual Genome dataset. The fact that many of these labels and relations are extremely rare in the Flickr30k dataset does not necessarily call their quality into question, as every dataset has different distributions. However, future work should look into removing/merging redundant labels and possibly adjusting the labels to the task, especially when working with a small label set. Concluding the discussion on categories and labels, it should be emphasized that even though there is room for improvement, the scene graph sufficiently covers many relevant categories and is thus suitable as a representation for the Flickr30k images.

3.2.2 Missing Information and Retrieval Methods

The scene graph is not yet an ideal representation of images. For instance, the scene graph does not always hold a suitable label for objects, there are not attributes for color, shape or size, and there is not representation for actions. In this subchapter, we will discuss which important information is not yet captured, thinking of the scene graph as a general representation of images for all kinds of tasks, not just for phrase grounding in particular. We also suggest various methods for retrieving the missing information and incorporating it into the scene graph. For instance, we discuss the limited label set and ways to retrieve more and better scene graph labels from the existing ones. We also examine how knowledge extracted from resources such as WordNet and ConceptNet could be included. We show how collective terms such as *group* or *couple* can add valuable information to the graph and suggest how to retrieve attributes and information about people's facial expressions and postures. Finally, we discuss how to detect action and scenes. Of course, the impact of newly added information differs strongly depending on the task, so we discuss in the Chapter 3.2.4 which information we consider relevant for phrase grounding.

Limited label set

The limited label set is one of the main issues that needs to be addressed in order to achieve a better representation. Many objects are not yet identified correctly or

precisely enough because there is no label for them. For instance, the scene graph can never accurately represent an image of a lion since there is no label for it. Most often, relevant objects are actually detected by the object detector but labeled wrongly or imprecisely. Thus, object detectors with a larger label set and possibly more specific labels will improve the scene graph representation substantially. Parcalabescu and Frank [2020] incorporate the *tfoid* object detector with 545 labels but with a low upper bound (50.0%), as well as *visgen* with 1600 fine-grained labels that achieves an upper bound of 87.9% [Yang, 2017]. Both object detectors are Faster R-CNN models trained on Open Images and Visual Genome, respectively. Note that a larger number of labels does not necessarily imply higher quality labels. We would have to analyse the new label sets and their distribution in order to determine whether they are suitable. While including a better object detector is the most straight-forward approach for increasing the scene graph quality, it is by far not the only one.

Inference from detected objects

Using the already detected objects, one can infer positions and labels of new objects without a detector, as can be demonstrated on the example of *body parts*. Body parts are very frequent and make up every fifth node in the scene graph. A qualitative evaluation, however, reveals that the complete body part set is hardly ever found. In many images only one ear, eye, leg or hand is represented, but not the other. Sometimes the missing body part is simply not visible in the picture as it is hidden by another object or cut off by the image frame. In other cases, however, the body part would have been detectable but it is not found nonetheless. Since we have knowledge about the number of body parts a person or animal normally has, the scene graph could be improved by systematically looking for supposedly missing body parts in areas of the picture where they are expected. Similarly, for relevant body part areas not represented by the scene graph label set, one could predict their location by composition or partial selection of already detected objects. For example, the *waist* or *hip* could be defined as the upper part of the pants, the *belly* as the lower part of the shirt, and the *chest* as the upper part (without arms). The body part *extremities* would consist of both arms and legs.

Controversly, we find many cases in which one person is assigned more than 2 eyes, ears or hands, and there are dogs with up to 6 legs and 3 tails. Figure 3.4 shows an example: the woman is assigned 3 eyes and 4 mouths, but the nose is not detected and *ear-2* is clearly in an impossible position considering *ear-1*. The issue of superfluous body parts could be resolved by implementing a maximum constraint for body parts as well as sanity checks for their positions: If *ear-1* is on the right

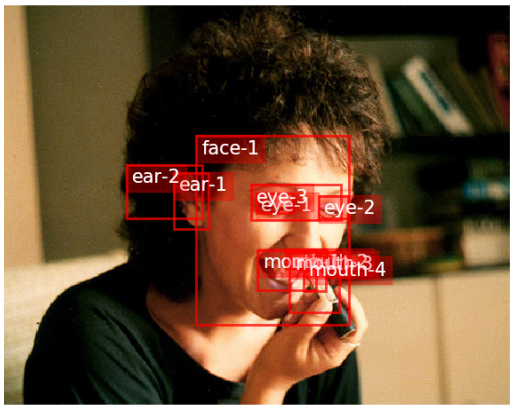


Figure 3.4: Parts of face.

Figure 3.5: Waves representing ocean.

side of the mouth, *ear-2* cannot possibly be on the right side as well (in frontal view). Also, since there is only one person, the objects *eye-3* and *mouth-2* through *mouth-4* have to be discarded. The same principle can be applied to other objects with many components that are often arranged in the same way, for instance cars. If 3 wheels are found, the position of the fourth can be inferred.

As a more general approach, one could infer new objects or scenes with the idea of *pars pro toto*, meaning *one part for the whole*. Sometimes parts of an object are represented in the scene graph but not the object itself. For example in Figure 3.5, there are 27 bounding boxes for *waves* but no box for *ocean* or *sea* or *water*. The numerous wave boxes, however, allow us to infer that there must be a body of water. This inference principle can be applied to many other objects and categories as well, e.g. wheels for a vehicle or bed for a bedroom. (Technically, body parts and clothes could also be considered *part of* a person but usually the person to whom the body parts and clothes belong is already represented in the scene graph.) The ConceptNet relations *PartOf* or *FoundAt* yield candidates for what the *whole* could be. The bounding box for the new object can be computed by merging all single bounding boxes when there are multiple *part* objects (e.g. waves), drawing a generous box (for instance 5 times as big) around the *part* object (e.g. wheel for vehicle) or simply considering the entire image as the new box for concepts of the type *scene* or *location* (e.g. bedroom).

WordNet and ConceptNet

Another way to improve and extend the label set is by directly injecting knowledge from resources such as WordNet and ConceptNet. From WordNet [Fellbaum, 1998] one could extract neighboring concepts such as synonyms, hypernyms and

hyponyms for each label in order to expand the label set. Each object would then be represented by several labels. While this may increase recall for some tasks, the multiple labels make the scene graph representation less consistent and precise. Parcalabescu and Frank [2020] include WordNet knowledge to contextualize their labels but rather than working with several labels they compute the mean embedding of the original label and the neighboring concepts. Thus, saving the word embedding vector rather than a set of labels could be an elegant way of incorporating several labels into the scene graph. Another idea is to add new edges to the scene graph for each neighboring concept, for instance the hypernym *animal* for *dog*. This would add structured contextual information to the scene graph and possibly aid many tasks, such as Visual Question Answering (*how many animals are in the image?*). Parcalabescu and Frank [2020] use the WordNet graph structure in order to compute an alternative similarity measure, named the WordNet path similarity.

ConceptNet [Speer et al., 2017] is a lexical knowledge graph that could be incorporated to extend the scene graph. ConceptNet contains a large vocabulary and provides more types of relations than WordNet so the concepts are more interconnected. However, it is less curated and "messier"[1]. The concepts are not represented in a hierarchical structure such as in WordNet. Furthermore, ConceptNet does not distinguish word senses apart from part-of-speech tags. ConceptNet could provide useful information such as what a certain concept is used for, by whom, where and in what way. Some relations (e.g. *FoundAt* or *UsedFor*) could help guessing scenes and actions shown in the image from the scene graph labels. However, these relations usually only provide useful results for some labels but not for many others. It is challenging to select which relationships to pursue for which types of concepts in order to generally improve the scene graph representation.

Attributes

It would be beneficial if the scene graph not only contained information about the location of an object but included a wide set of attributes, for instance color, size, form and material. In a caption, this information is often expressed by an adjective, e.g. *green jacket*, *big dog*, *wooden chair*.

The color can be retrieved by inspecting the color of the pixels in the respective bounding box. One could assume that the most frequent pixel color in the bounding box is the color of the object captured by it or pursue a more sophisticated approach as described in Chapter 4.4. For patterns, one could train a classifier that can

[1] `https://github.com/commonsense/conceptnet5/wiki/FAQ#comparisons-to-other-projects` (Last accessed: 21.09.2020)

tell apart basic patterns such as striped, dotted or checked. ImageNet provides a collection of 10k images, labeled with information for the attributes color, pattern, shape and texture that could be used for training [Deng et al., 2009]. If more data is needed, one can also extract ground truths from Flickr30k captions by inspecting the attributive adjectives. Since many adjectives are relative, one would have to compare the quality of the surrounding objects as well, for instance, in order to determine whether or not something is small. Perspective would also have to be considered since objects in the background naturally have smaller bounding boxes than those in the foreground, even if the their actual size is similar. In general, attributes are essential for differentiating objects with the same label, for instance a blue shirt and a red shirt. The attribute can be the only distinctive characteristic. The original scene graph only provides the position and (through the relations) neighboring objects as attributes.

The attributes can either be managed as an external list of attribute keys and values for each object, or they can be integrated into the scene graph by adding a new node that is connected to the object in question. The node label would be the attribute's value (e.g. *red*) and the edge label would indicate the attribute's name (e.g. *color*). Attribute nodes are of a different type than object nodes as they do not represent an object in the image and have no coordinates.

Collective terms and plurals

As mentioned above, the scene graph generally captures single entities and does not provide labels for collective terms, such as *crowd*, *group*, *couple*, or plurals in general. The only plural labels the object detector generates are *men* and *people*, although these nodes do not necessarily sum up all *man* or *person* objects. The remaining labels are in singular form, even in cases where the plural would be correct, for example *jean* rather than *jeans* or *short* rather than *shorts*.

Since many images contain several people, animals or objects of the same type, having a way of representing plurals or groups of objects in the scene graph is crucial. The simplest way of adding plural nodes to the scene graph is by grouping all objects with the same label together, creating a new bounding box by merging the single boxes, and labeling the new box with a plural tag. The original single bounding boxes are not removed or altered. In a more sophisticated approach, one could generate plural boxes for all kinds of object combinations, for instance, *boy-1* and *girl-1* can be summarized as *children* while *person-1* through *person-9* receive a plural box named *crowd*. ConceptNet could help automatically select a suitable label, for instance with the *PartOf* relation: *sheep* $\rightarrow$ *PartOf* $\rightarrow$ *flock* or *student* $\rightarrow$

(a) Standing. (b) Sitting. (c) Walking. (d) Exercising.

Figure 3.6: Postures.

PartOf → *class*. This does not yield reasonable output for all labels, though: *man* → *PartOf* → *British Isles*, so careful selection is necessary to avoid noise and errors.

Facial expression and posture

Since many Flickr30k images contain people, it would be helpful if the scene graph contained some information about their facial expression and posture. The facial expression tells us something about the mood or emotional state: a person can be smiling, laughing, crying or frowning; their face can express joy, anger, surprise, sadness, disgust or fear, among many other emotions. The emotional state gives valuable information about the scene depicted in an image. The scene graph provides bounding boxes for mouth, face and eyes, so one could feed these sections of the image into a ready-to-use facial expression recognition system, for instance *Deep-Emotion* that uses an attentional convolutional network [Minaee and Abdolrashidi, 2019]. Alternatively, one could train a classifier using the face objects and the emotions mentioned in the caption as training data. While not directly a facial expression, the direction in which a person is looking can also be a valuable attribute, which can be inferred by the position and orientation of the eyes and face.

Similarly to facial expressions, the physical configuration or posture of a person can contain information about what the person is doing, for instance standing, sitting, kneeling, squatting, lying, walking, running or jumping. One could exploit the fact that usually a large number of body parts and clothing items are detected for each person. The arrangement and distribution of these objects could be used as features to predict the posture. For instance, if head, shirt, pants are arranged in a vertical line, the person is standing (see Figure 3.6a). For a crouching person or person sitting on the floor, the upper body (shirt/jacket) and the legs (pants) are in a

horizontal line rather than vertical, possibly partially overlapping (see Figure 3.6b). For standing people, the scene graph usually does not include both legs as they cannot be clearly separated and identified. Therefore, if both legs are captured by the scene graph and they are not highly overlapping, the person is walking (see Figure 3.6c). If the legs are even clearly apart, the person is running. Unusual arrangements of clothes and body parts (e.g. one leg at same level as arm) can be an indicator for high-movement actions such as exercise (see Figure 3.6d), although it should be pointed out that for highly acrobatic postures (both legs in nearly horizontal position or foot higher than head), the detector struggles at differentiating arms and legs.

Since many actions have characteristic body postures one could train a classifier for learning them, possibly in combination with surrounding objects: a person lying on a bed or couch is probably resting or sleeping, a person sitting at a table with food is probably eating, a person sitting cross-legged on the floor with his eyes shut is probably meditating, a person carrying a backpack walking through an area with many trees is probably hiking, and several children running after a ball are probably playing a ball game. Andriluka et al. [2014] created a dataset for training and evaluating human pose estimation from 2D images. It also contains activity labels for each of the 25k images, which allows learning of pose and activity in context of each other.

Actions

Actions or activities are not represented in the scene graph at all, besides a few relations such as *wearing* or *holding*, which are generally more descriptive of the person rather than the actual action. Object detection only solves half the task of image analysis. The objects have to be put in context with each other so that the scene and actions shown in it can be inferred. The scene graph makes a start by linking objects by their visual relations and thus provides more structured information than a simple list of objects. However, a complete scene graph would contain action attributes and relations, for instance DOG-1*: action = run* or GIRL-1 *rides* BICYCLE-2.

Action recognition from still images is a well-established problem with several benchmarks [Yao et al., 2011; Andriluka et al., 2014; Chao et al., 2015]. One could employ an existing action detector or train an action classifier using the detected scene graph objects, their frequency, constellation and relations as features. The ground truth actions (verbs) can easily be extracted from the Flickr30k captions. Additional resources such as ConceptNet, WordNet or just plain text corpora could be used to

learn which (combination of) objects co-occur with which actions, in order to reduce the number of candidate actions to select from. For instance *children* and *ball* frequently co-occur with the action *play*, while *man* and *computer* and *desk* are typical for the action *work*. As actions are often complex and involve several objects, one could employ FrameNet for finding all relevant *frame elements* and linking them to each other [Baker et al., 1998; Ruppenhofer et al., 2006]. For a scene in which a boy hits a baseball, the frame would be *hit* and the core frame elements would be agent (*boy*) and target (*ball*). Non-core frame elements include instrument (*baseball bat*), manner, means, place, purpose and time. Since the scene graph only represents objects, one would have to select frame elements of the semantic type *physical entity* (i.e. instrument). Silberer and Pinkal [2018] tackle the task of visual semantic role labeling with a model that grounds semantic roles of a frame in image regions.

Actions are essential for the representation of an image since they bring a *story* to the image. Images of various actions can have the same main objects in them, for instance two men, but the story they tell is completely different: Two men can be sitting together on the couch watching TV or running through a park or eating together at a restaurant or working on a construction site. The action can be inferred by their posture (sitting, running), the objects they interact with (couch, table, tool) and most importantly by the scene (living room, park, restaurant, construction site). While the scene does not necessarily define the action, it narrows down the candidate actions. In a restaurant, one can eat, drink, talk, celebrate or have a business meeting but one probably cannot ride a horse or take a bath. Again, ConceptNet can suggest possible candidates, for instance *kitchen* $\rightarrow$ *UsedFor* $\rightarrow$ *cooking / baking a pie / eating meals*. However, in order to use the scene as a means of identifying actions, the scene first has to be identified.

Scenes

The category *scene* is special since it does not describe a specific object but the place or scenario in which the people and objects in the image are shown. The scene graph does not include any *scenes* in the label set so they are not explicitly represented in the scene graph. Identifying the scene is challenging as it not only requires object detection but also identification of the interaction of these objects and their actions. Just based on the pixels, three images of a dinner party can look very different. What makes it a dinner party are the people (and the fact that they are dressed up), the location (living/dining room), a set table, nice tableware, fancy food and drinks, the people talking to each other, eating, drinking, celebrating.

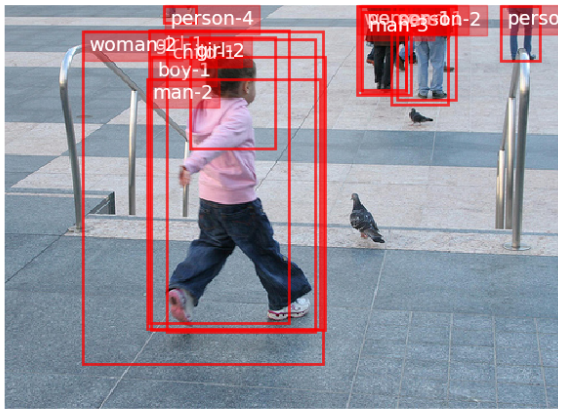


Figure 3.7: Multiple boxes for person.

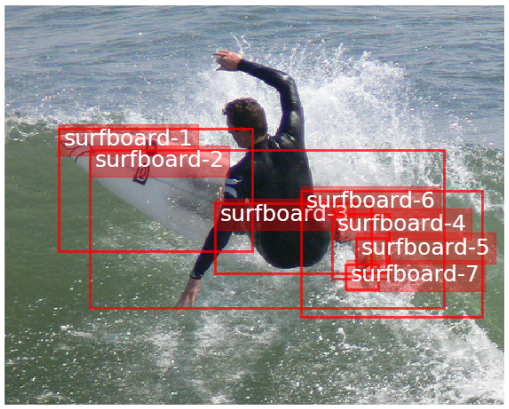


Figure 3.8: Multiple boxes for object.

In general, identifying the scene is crucial for analyzing and interpreting an image as it offers a context for all objects. Depending on the scene, an object can have different functions: a woman holding a racket on a tennis court is a tennis player, while the same woman can be a driver when shown at a steering wheel of car, or a guest when depicted at a dinner party. As mentioned above, the scene and action are often linked: knowing the scene helps in identifying the action, and vice versa.

The *pars pro toto* method may help identify some scenes, for instance *waves* for *ocean* or *beach*. However, smaller items in the background are often not recognized (correctly) and thus cannot be used to guess the scene using ConceptNet or other sources. An alternative approach would be to train a classifier to differentiate the different (types of) scenes and backgrounds based on the scene graph objects (and their relations) in the picture. The ground truths can be extracted from the *scene* category in the captions. When focusing on the background, one could first eliminate all elements in the foreground. This can be done by applying an image measure to guess whether an object is in the foreground (size, scene graph degree) or by removing all objects mentioned in the caption. Then one can cluster the backgrounds and determine which scenes often have similar backgrounds.

3.2.3 Multiple Representations

This section does not cover missing elements in the scene graph but superfluous ones. When looking at any scene graph representation, one can observe that many people are captured by several bounding boxes, most often with differing labels. For instance in Figure 3.7, the little girl in the foreground is captured by six different bounding boxes. They are labeled *girl-1*, *girl-2*, *child-1*, *woman-2*, *boy-1* and *man-2*.

The bounding boxes are mostly overlapping, some boxes being precise while others are clearly too small or too large. This happens with nearly every image and can be observed even if only the 20 most confident triples are considered to create the scene graph (in this case, *girl-1*, *child-1* and *boy-1*), which suggests that using the full set of triples is not the source of the problem. While the phenomenon is most prevalent for people (possibly because of the large set of people labels), it can be observed for many objects: Figure 3.8 shows that there are seven different scene graph objects for the one surfboard, each with a different bounding box. For non-people objects the labels are usually the same (e.g. surfboard), although multiple semantically similar labels can be found as well, for instance *truck* and *car*. This effect is probably caused by the detector trying to identify as many objects as possible, at the cost of finding some objects multiple times.

Depending on the task, the multiple bounding boxes for the same object do not directly pose an issue but the overall quality of the image representation clearly degrades since ideally every object should be represented by exactly one scene graph object and bounding box. For tasks such as visual question answering, this phenomenon is an obvious source of error: if asked for the number of little girls or surfboards in an image, this representation will not yield the correct answer. Discarding superfluous boxes or merging them is required to remedy this flaw and generate a clean scene graph representation. We will address this issue in more detail in Chapter 4.3.

3.2.4 Relevance for Phrase Grounding

The goal of the scene graph is to accurately and generally represent an image so it can be used for any task involving vision and language. Above we discussed what information is still missing for a complete representation. However, depending on the task to be solved, not all information is required or of equal importance. In this section, we will review the information present or missing in the scene graph and assess whether it is relevant for the phrase grounding task.

Categories and Labels. We discussed the Flickr30k coverage of the categories in Table 3.1. It suggests that *people*, *clothes*, *animals* and *vehicles* are covered sufficiently while *scenes* and *instruments* are not represented at all. The category *other* needs to be extended with a larger number of labels, while for the *body parts* it would help to retrieve more area-based body part descriptions. In general, a greater number of precise labels would naturally help with phrase grounding but the given objects and their labels are sufficient to correctly align up to 65% of all phrases, even 83% if label category and phrase category do not have to match.

Body parts. Completing the body parts set for every person (or animal) is not necessary for successful phrase grounding. While it may help for other tasks or indirectly aid retrieval of other information types, for instance posture and action, there is not much to gain in terms of phrase grounding. Searching for missing body parts could even introduce errors and noise. However, inferring areal body parts such as *waist* or *chest* from already detected body parts and clothes could improve the system as it is simple, relies on already detected objects and resolves a shortcoming.

Pars pro Toto. Deriving new objects with the *pars pro toto* principle is in general an interesting way of extending the scene graph. However, it is not the most straight-forward or promising method as it depends on the quality of the ConceptNet suggestions and on correctly guessed bounding boxes. As we seem to have a sufficient number of bounding boxes representing the phrases in Flickr30k but not always suitable labels, the focus should be put on precision (more correct labels) rather than recall (more boxes), which renders this method unsuitable.

WordNet and ConceptNet. Parcalabescu and Frank [2020] already showed that WordNet can be used to inject knowledge into the scene graph and improve phrase grounding. We also mentioned several applications in which ConceptNet knowledge may be used to infer a new object, scene or action. However, while ConceptNet look-ups yield perfect results in some cases, they may yield no result or – even worse – unsuitable ones in others. These can confound the phrase grounding system and may degrade performance to a greater extent than correct look-ups could improve it. In contrast to WordNet, ConceptNet does not provide a set of relations that exists for every concept, such as hypernyms, hyponyms or synonyms. Every concept has its own set of relations so it is difficult to find general rules for extracting related concepts that can be applied to all scene graph labels. The fact that the labels are very general and ConceptNet does not provide synsets for word sense disambiguation presents another challenge. For all these reasons, it is difficult to exploit ConceptNet for directly extending the scene graph in its static representation. However, ConceptNet can improve phrase grounding by finding a connection between scene graph label and caption phrase and thus provide a more versatile extension, as we will discuss in Chapter 5.

Attributes. As long as objects can be clearly differentiated by their scene graph label, attributes do not have a big impact on the phrase grounding task. However, attributes become essential as soon as there is more than one scene graph object with the same label that could match a candidate phrase. For instance, a blue shirt and a red shirt can only be differentiated by their color as the label *shirt* is the same for both objects. A manual analysis of 100 captions revealed that 60% of the mentioned attributes are colors and 19% are sizes, although the size is never used

to discriminate from other possible candidates (e.g. *little girl* when there is no *big girl*). Material/texture, pattern and shape are relatively rare. Therefore, color is - next to position - the most informative attribute for phrase grounding.

Collective terms and plurals. The missing distinction between singular and plural objects and the lack of object groups representing several single objects clearly affects the representational quality of the scene graph. The Flickr30k coverage increases by 7% when plural phrases are excluded, which strongly suggests that they are not represented well. For the task of phrase grounding, which includes both singular and plural phrases as well as collective terms, the linguistic number of the phrase must be considered and the option of assigning to a collection of objects must be enabled. We will discuss the scene graph enhancement with plural objects and the distinction between singular and plural phrases in Chapter 4.2.

Facial expression and posture. For general representation, it is important to have information about the facial expression, emotions and the posture of a person depicted in an image. However, as these features are typically described with adjectives or verbs in a caption (e.g. *a happy girl* or *a boy crying* or *a man walking*), this information does usually not directly aid phrase grounding. In some cases they may help differente two people (*the laughing girl* vs. *the running girl*) but most often this type of information does not affect the phrase grounding performance.

Actions. A similar conclusion can be drawn for actions: they are usually expressed with verbs – and phrase grounding concerns itself mostly with nouns. However, the identification of the action indirectly helps phrase grounding when it allows specifying the function of a person. In captions, people are sometimes not referred to by a general term (e.g. girl, boy, woman, man) but by their profession or role, for instance *construction worker*, *hiker*, *student*, *cook*, *doctor* or *patient*. So since the action (working at construction site, hiking, studying, cooking, tending to patient or being treated) enables inferring the role of a person in an image, the actions provide relevant information for phrase grounding. When annotated with frame elements, grounding one phrase may even have cascading effect. The image in Figure 3.9 shows an example: if the action *give* is inferred from the image, the frame elements donor (*woman-1*), recipient (*woman-2*) and theme (*toy-1*) can be linked within the scene graph. Since the caption *A young woman gives a doll to an older woman* also includes the frame *give* and the phrases can be assigned to frame elements, correctly grounding one phrase (e.g. *a young woman* to *woman-1*) entails the groundings for the other phrases.

Although rare, in some cases the action attributed to a person in a caption is the only clue for correctly grounding that phrase. Figure 3.10 and two of its captions give an

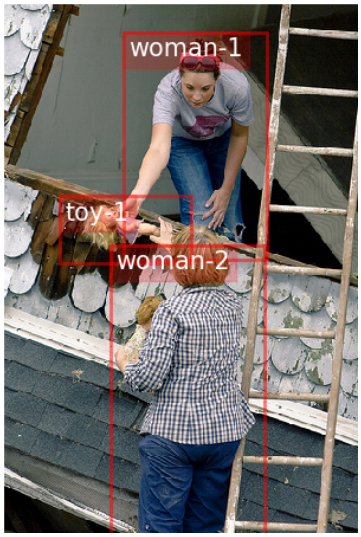


Figure 3.9: Frame elements for *give*.

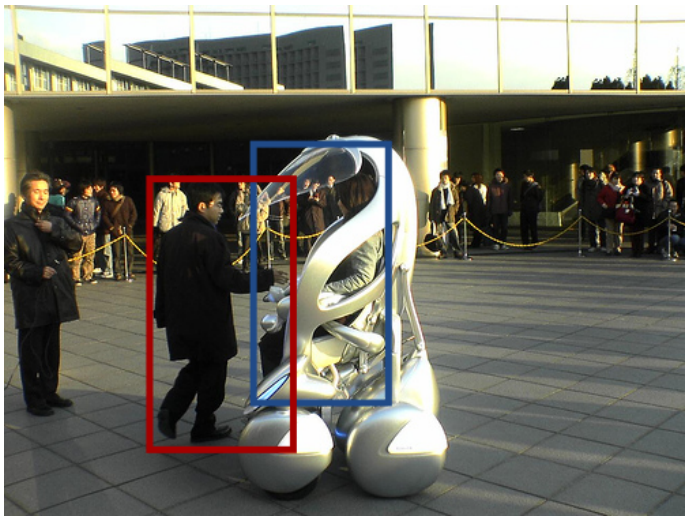


Figure 3.10: *Walking* vs. *riding in*.

example. The structure of the two captions is very similar (person+verb+vehicle), and both describe only one person – but not the same one. Caption 2 describes the person in the vehicle (*riding in*) in the blue bounding box and Caption 3 describes the person next to the vehicle (*walking up to*) in the red bounding box. Without considering the verbs, this phrase grounding problem cannot be resolved.

Caption 2: A person [#ENT363] **riding in** a futuristic single-person vehicle.
Caption 3: Man [#ENT361] **walking up to** silver 4-wheeled chair.

Scenes and backgrounds. While there are scene phrases in the captions, many of them have no bounding box assigned, so they are excluded from the evaluation and do not have an effect on phrase grounding performance. However, similar to actions, scene identification helps specify the people and objects in the image. A tennis court makes the person with the racket a tennis player, a ballet studio makes the girls jumping up in the air ballerinas, a concert makes the people with instruments musicians or a band and the crowd the audience or fans. Objects can also be specified more easily when put into the context of a scene: a pole (which is a scene graph label) can represent a street lantern when found on a road, a hockey stick in context of a hockey match, a railing when found on a ship, balcony or lookout point, a broomstick in the context of cleaning (or a Quidditch match), or a fishing rod at a lake or other body of water. Thus, the context or scene can extend the limited label set with more precise labels.

Multiple Representations. The fact that many people and objects are represented by several boxes does – at first – not seem to pose an imminent problem

since the bounding boxes are mostly overlapping and for the performance of phrase grounding it does not matter which of the boxes is chosen. Multiple boxes may even increase the chance of finding the correct alignment since the boxes often have differing labels, especially for people as seen in Figure 3.7, and this increases recall. However, there are harmful consequences of this phenomenon as well. First of all, multiple labels reduce precision: the girl in Figure 3.7 receives the labels *girl*, *child*, *woman*, *boy* and *man*. While adding *child* may improve phrase grounding for captions that describe the girl as a child and while the label *boy* cannot be entirely discarded since the gender of the child can be hard to determine with absolute certainty, the labels *woman* and especially *man* are clearly wrong. With this annotation, the girl could easily be aligned to the phrase *an old man*. The multiple boxes raise another problem when combined with the merging method used in some phrase grounding systems, as will be discussed in the next subchapter. Therefore, superfluous boxes not only impair phrase grounding but also compromise the quality and reliability of the scene graph by introducing noise and errors.

3.3 Phrase Grounding using the Scene Graph

Having discussed the quality and capabilities of the scene graph, we will now evaluate its performance on the phrase grounding task. We employ a simplified version of the system introduced by Parcalabescu and Frank [2020], omitting the incorporation of scene graph relationships, knowledge injection through WordNet and Open Images, as well the path similarity measure. This makes it easier to study the individual effects of our enhancements. For each image/caption pair, phrases and scene graph object labels are embedded using 300-dimensional word2vec embeddings [Mikolov et al., 2013]. For each embedded phrase and candidate label, the cosine similarity is computed. The label with the highest similarity is chosen as the best candidate and the phrase is aligned to the object's bounding box. If there are several objects with this label, the largest is selected. If the IoU of the ground truth and predicted box is higher than 0.5, the phrase grounding is considered successful. Since the system aligns caption phrases to scene graph objects (and their bounding box), we refer to it as a phrase to scene graph object aligner, or simply aligner.

3.3.1 Performance and Analysis

The test set consists of 10k images with 50k captions, including 171k phrases. The overall accuracy is 47.9%, respresenting that for 47.9% of all phrases the correct bounding box is predicted. Figure 3.11 shows the distribution and performance

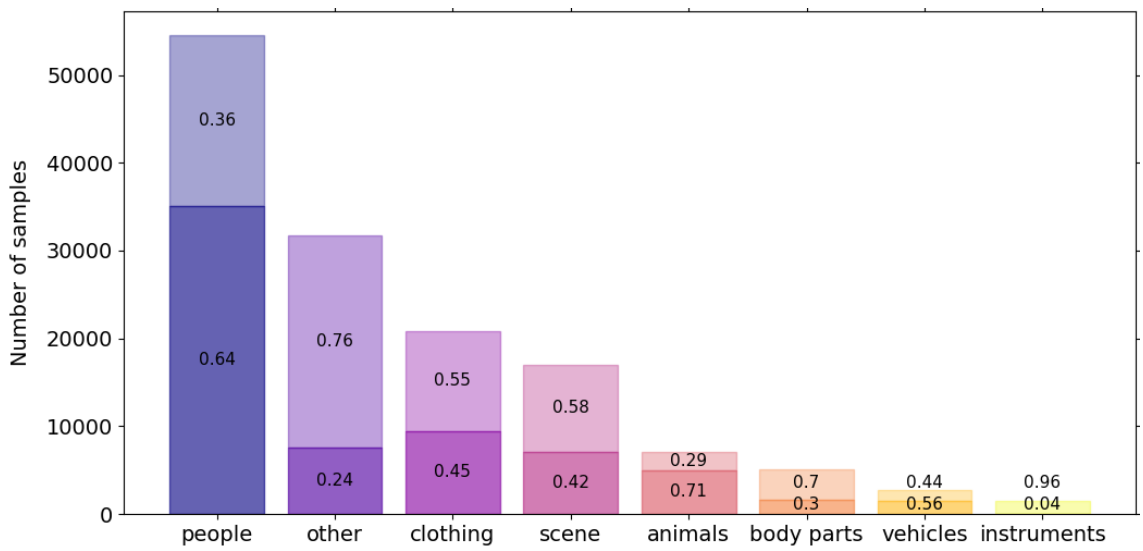


Figure 3.11: Phrase grounding accuracy by category.

for the different Flickr30k categories, the darker areas representing the successful alignments. The performance highly correlates with the Flick30k coverage without category constraints in Table 3.1. The *animals* category performs best with 71% correct groundings, followed by *people*, *vehicles* and *clothing*; all four categories have Flickr30k coverage higher than 90%. As for *body parts* and *other*, only every third or fourth phrase is grounded correctly. *Scenes* and *instruments* do not have explicit scene graph labels; but *scenes* has a phrase grounding accuracy almost 10 times higher than *instruments*, probably because some *scenes* describe visual objects such as *building* or *street*, which are detectable (albeit with another label), while *instruments* are often not represented by any bounding box at all. From the correctly aligned phrases, 10.5% are *scenes* and nearly all of them are aligned to an object from the category *other*. Interestingly, in most cases with incorrect groundings the predicted bounding box is of the category *other*. This category is very broad so it is not surprising that if there is no obvious match, the highest similarity would most likely be computed for one of the labels in this category.

Performance varies considerably over the different images and captions. For 19.3% of all captions no phrase can be grounded correctly and in 14.7% of all captions, all of them are aligned correctly. On the image level, 1.2% of all images have all their entities over all 5 captions perfectly aligned, while 3.5% have no entities aligned correctly at all. For most images, the percentage of correct phrase alignments lies between 30% to 70%.

If the phrase contains the label of the correct scene graph object, the alignment is usually successful, for instance *a man* (rather than *a construction worker*) for *man-1*, or *a hat* (rather than *a beret*) for *hat-2*. In general, the alignment is more successful if basic and general language is used instead of specific terminology (*dog* vs. *terrier mix*) since the phrase and label are more similar to begin with. Abstract descriptions and interpretations make alignment more difficult, for instance if the man in Figure 3.8 is not described as a *man* but as a *surfer*. Other examples for abstractions and roles are *worker, player, team, opponent, teenagers, students, friend, daughter, orchestra, security guard, climbers* or *performers*. Some roles could be determined by inspecting the clothes, for instance a police uniform for a police man. However, there are many roles for which the looks of the depicted object or person strongly depend on the context – for example what a *player* looks like varies with the game or sport in question. Terms such as *friend*, *daughter* or *opponent* depend on the relation of the people in the image and thus cannot be inferred by their looks alone. Finally, some of these phrases are not even aligned to a person object, let alone the correct one. In fact, the cosine similarity for word embeddings does not always promote candidates from the phrase category, so the phrase *student* (category: people) could be aligned to *computer* (category: other) since it is ranked higher than *man* (category: people). As another example, the phrase *construction worker* is aligned to *building* rather than *person* or *man*. As a final observation, 71% of the predicted bounding boxes for failed samples are too small. This is not surprising since the Flickr30k ground truth boxes are on average three times larger than the single scene graph objects.

As mentioned above, the scene graph only represents single objects. Since the phrases often describe multiple objects, it is important to include a way to represent multiple objects in one bounding box. Some phrase grounding systems, including the ones by Wang and Specia [2019] and Parcalabescu and Frank [2020], pursue the following method: if there are several objects for the highest ranking label, the union of their bounding boxes is predicted. Note that Wang and Specia [2019] and Parcalabescu and Frank [2020] do not explicitly include this union box method as a means for handling plural phrases but as a tie-breaking strategy for objects with the same similarity score, although they mention that it can help alinging phrases representing multiple instances.

While without the union boxes 71% of all incorrectly predicted boxes are too small, this method reduces it to 36%. On average, there are 14 labels per image that appear several times and thus could potentially generate union box. Figure 3.12 and two of its captions provide an example: the blue union box covers both dogs and thus the phrase *two dogs* in Caption 4 can be correctly grounded. The downside

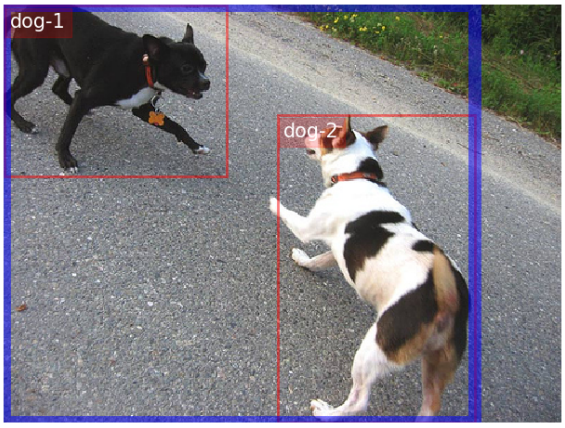


Figure 3.12: Union box for dogs.

Figure 3.13: Union box for hands.

of this method is that it eliminates the option of assigning a singular phrase to a single object whenever there are several objects with the same label. For instance, *a black dog* in Caption 5 can never be aligned to *dog-1* since the two dog objects are automatically merged. While this merging method enables the alignment of plural phrases, it reduces the accuracy for phrases representing only one object. Therefore, the overall phrase grounding accuracy only increases slightly, from 47.9% to 48.1%.

Caption 4: **Two dogs** on pavement moving toward each other.
Caption 5: **A black dog** and **a spotted dog** are fighting.

Combined with the frequent issue of having multiple boxes for a single person or object, merging boxes with the same label causes even more mistakes. People often receive many different and sometimes unsuitable labels (e.g. wrong gender), so when union boxes are generated using these incorrect labels, the union boxes end up being too large. The girl in Figure 3.7, for instance, is wrongly labeled as *man-2*. When grounding the phrase *the men in the background*, the only choice is the union box for *man*, which correctly includes *man-1* and *man-3* in the background and unfortunately also *man-2* (the girl). This box would cover half the image and clearly does not represent *the men in the background* accurately. Even if the label remains the same, multiple boxes for the same object can result in overly large union boxes. For the surfboard in Figure 3.8, the box surrounding *surfboard-1* through *surfboard-7* is so large that it is not recognized as the correct solution when comparing to the ground truth, while *surfboard-1* on its own would be considered correct.

The union boxes also increase the number of cases in which the correct bounding box is found but for the wrong reasons, as demonstrated by Figure 3.13. The guitar is not represented in the scene graph, as it often happens with instruments. For

the phrase *a guitar*, the label with the highest cosine similarity happens to be *hand*. Without the union box method, no matter which hand is selected, the IoU score does not confirm a match. When the union box for *hand* is added, however, it roughly covers the area of the ground truth box for guitar – and the guitar is considered as correctly aligned. The label does not fit at all but the box (coincidentally) does. The introduction of union boxes increases such cases drastically since they are naturally larger than the single object boxes, which increases the change of high IoU with the ground truth boxes. This phenomenon prompts us to critically examine the evaluation metric for phrase grounding.

3.3.2 Evaluation Metric

Phrase grounding is generally evaluated by comparing the predicted bounding box to the ground truth box. The bounding box is given by the selected scene graph object. The ground truth is provided by the Flickr30k Entities dataset, which contains gold bounding boxes for all entities. The bounding boxes were manually drawn by multiple annotators so most entities have several boxes, which are usually quite similar but not completely identical. To create a gold bounding box, these boxes are merged into one bounding box that includes all single boxes. Note that this process makes gold bounding boxes slightly larger than they should be.

If the IoU score of these predicted and gold bounding box is greater than a specific value (normally 0.5), the grounding is considered correct. While this is a simple and fast evaluation metric, it disregards how the predicted bounding box is generated, what label is assigned to it and whether it semantically matches the phrase. As long as phrase grounding is reasonably accurate and correct labels are not required, this may not matter for most applications. However, when evaluating and improving phrase grounding systems it is important to understand that this metric does not reflect how many phrases were aligned correctly by chance and that many labels of detected objects may not be correct, even if the selected image region mostly is.

Since the scene graph has a limited label set, one has to differentiate between cases in which the right object is aligned but an unsuitable label is chosen due to lack of a better one, and cases in which the wrong object was aligned but the label happens to be the best candidate and the box matches. Figure 3.14 gives an example for the first case: the plush lion is recognized as an animal-like object but for lack of a better option, it is assigned the label *bear* – which is the optimal solution given the label set. The other case is represented by Figure 3.15: the phrase *small child* is correctly grounded by aligning it to *child-1* but the phrase *red ropes* is assigned to the box of *girl-1* (again a superfluous box). While the IoU score classifies it as a

Figure 3.14: Imprecise label.

Figure 3.15: Incorrect label (*girl* for *ropes*).

match, the label betrays that the red ropes were not actually detected and it is thus incorrectly counted as a successful alignment. The ground truth box of *red ropes* covers the entire image which reveals how forgiving the evaluation by IoU can be.

Extensive manual evaluation would be required to determine exactly how frequently this phenomenon occurs so we compute a rough estimate. Among the phrases that are considered correctly grounded, for 32.0% the predicted label is not included in the phrase and for 15.9% the predicted category and phrase category do not match (for 8.8% both holds true). Of course, this does not necessarily mean that all those phrases were only accidentally correctly aligned, as the word embeddings are meant to build a bridge between phrase and label, even if they are not the same word or category. However, these numbers clearly show that label or category do not have to match in order for an alignment to be considered correct.

When extending the scene graph and improving phrase grounding, we use this evaluation metric as a general guideline for measuring performance. One should bear in mind, however, that this metric does not necessarily reflect the quality of the scene graph or even the exact accuracy of the phrase grounding system. Some of our enhancements will qualitatively improve the scene graph and phrase grounding system but not result in any performance increase as it affects aspects that are not covered by this evaluation metric, for instance the object reduction to remove multiple representations discussed in Chapter 4.3. Since we use a fully transparent system, we can reconstruct precisely how the predicted box was generated and selected, which allows us to examine whether the evaluation method provides fair assessments.

4 Scene Graph and Aligner Enhancements

In this chapter, we discuss the implemented modifications to the scene graph with the aim of improving its quality as a structured representation of image content. The scene graph is enhanced by addings plural objects and attributes and by removing superfluous objects. We use phrase grounding to evaluate the impact of our enhancements on a real-world task. We also implement some optional extensions for the aligner that filter out or promote alignment candidates depending on characteristics of the phrase and the scene graph objects.

We generate and discuss three different scene graph (SG) representations. The representation $\text{SG}_{original+attr}$ is mostly identical to the original scene graph representation as described in Chapter 3.2, with the addition of attribute edges for color and background (see Chapter 4.4 and 4.5). As the extensions that make use of the color and background attributes can easily be deactivated, there is no need to keep a separate version without the attributes. The representation named $\text{SG}_{attr+plurals}$ contains all the nodes and edges from $\text{SG}_{original+attr}$ but is extended with nodes representing plural objects (see Chapter 4.2). The final representation $\text{SG}_{attr+plurals+red}$ is based on $\text{SG}_{attr+plurals}$ but with 3% fewer nodes due to an object reduction method that removes multiple representations of the same object (see Chapter 4.3).

The three main aligner extensions consist of number distinction for phrases and objects (see Chapter 4.2), inclusion of the color attribute (see Chapter 4.4) and grounding assistance for people objects (see Chapter 4.5). We report the results and their statistical significance for all three scene graph representations and the aligner extensions (both separate and combined) in Table 4.2. The columns capture the scene graph representations and the rows the aligner extensions. The scores represent the phrase grounding accuracy, evaluated on a test set of 172k phrases, derived from 50k captions describing 10k images. The version labeled *vanilla* refers to the aligner without any extensions. The baseline is computed by predicting the entire image as the aligned bounding box for each phrase.

4.1 Correction of Misspellings

We noticed that roughly 1% of all phrases receive a word embedding vector consisting solely of zeros because none of the words in the phrase can be found in the pre-trained word embeddings. This is often caused by misspellings in the captions (e.g. *trumped* instead of *trumpet*). For phrases with no successful word embeddings look-up, we try to auto-correct the phrase using the python library *autocorrect*[1]. Among the 1770 phrases with no embeddings, auto-correction resolves the problem for 494 phrases (27%) and pushes the phrase grounding accuracy from 18.5% to 32.99%. On the complete test set, accuracy improves by 0.05%, which is not negligible considering that this is an extremely minor fix affecting only 0.29% of all samples. Since auto-correction for phrases with no embeddings is a general and safe enhancement, we include it in the vanilla aligner.

4.2 Plural Objects

As discussed in Chapter 3.3.1, the scene graph only represents single objects, which is problematic for grounding phrases that represent several objects. Many systems follow the simple method of merging the bounding boxes for all objects with the same highest ranking label, which can result in the desired plural bounding box but eliminates the option of aligning a phrase to a single object. The downsides of this uncontrolled merging approach were discussed in detail in Chapter 3.3.1. We introduce a slightly different yet simple and effective way to process both singular and plural phrases.

Rather than merging the boxes with the highest ranking label at grounding time, we generate all possible plural boxes and integrate them into the scene graph before phrase grounding, as this information could be generally useful for other tasks as well. This also saves processing time during phrase grounding, which can be run repeatedly in different configurations without having to re-compute plural boxes each time.

To generate plural objects, the bounding boxes for all objects of the same label are merged, resulting in a bounding box that covers all objects of that label. The merged box is computed by selecting the lowest value of all *xmin* and *ymin* coordinates and the highest of all *xmax* and *ymax* coordinates. Since the resulting box is a still a square, it can have an area that is much larger than the sum of all individual areas since the individual objects may be widely distributed across the image (see left top

[1] `https://pypi.org/project/autocorrect/` (Last accessed: 21.09.2020)

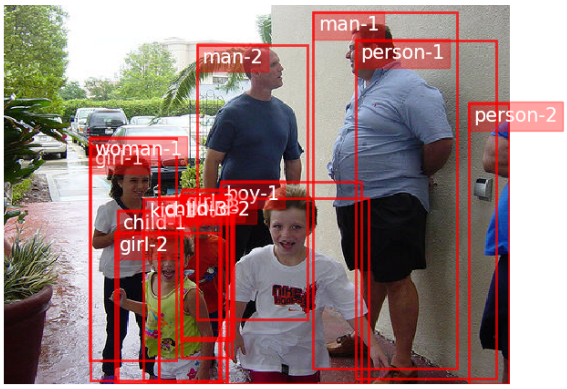


Figure 4.1: Single objects.

Figure 4.2: Plural objects.

corner of *people-pl* box in Figure 4.2). As the ground truth box for plural phrases is computed exactly the same way, this does not present an issue, although in future work a more precise way of merging of objects could be advantageous, especially when working with segmentation masks rather than bounding boxes. The new bounding box represents several objects and so rather than an object ID, it receives the tag *pl*, as seen in Figure 4.2. The original single objects are not removed or altered so this new scene graph representation contains singular objects as well as collective plural objects for all present labels.

For objects that represent people, we adjust this procedure to account for the fact that people labels are semantically overlapping. Besides generating a plural object for each person label (e.g. *woman-pl* for *woman-1* and *woman-2*), we also group some labels together to arrive at more general collections: *female* (woman, girl, lady), *male* (man, boy, guy), *children* (child, kid) and *people* (woman, girl, lady, man, boy, guy, child, kid, person). Figures 4.1 and 4.2 show examples for singular objects and plural objects generated in this way. For instance, the box *man-pl* is generated by merging *man-1* and *man-2* while the collection box *people-pl* covers all people of any label in the image. The three children in the yellow, orange and white shirt all have the label *girl* (among others) so they are surrounded by the plural box *girl-pl*. The two children in the front are labeled *child* and *kid* so their combined box is labeled *children-pl*. Ideally, the other two children would be included in that box but since they do not have a *child* or *kid* label they are excluded; the label *girl* is not sufficient to be classified as a *child* since it could refer to an adult woman.

In fact, the ambiguous use of terms such as *girl* or *boy* generally presents a challenge: *girl* can be used to describe a female child but it can also refer to a teenager or young adult woman. The same can be observed for *boy*, although less frequently. Thus, summarizing a *girl* and a *boy* (or a *girl* and a *child*) as *children* is often incorrect,

so we exclude this option. This eliminates the opportunity of creating a children box that includes all four children in Figure 4.2 but prevents incorrect children boxes in many other images. In captions, the adjectives *little* or *small* are often used to emphasize that the described people are children, for instance *a small boy* or *a little girl*. When both children and grown-ups are shown in the image, the terms *girl* and *boy* are usually exclusively used for children. However, we do not incorporate this insight in the plural object generation, since we aim to keep scene graph representations independent from the captions and hence useful beyond phrase grounding. However, as an aligner extension we promote phrases such as *little girl* or *small boy* to be aligned to scene graph objects labeled *girl* and *boy* rather than *woman* and *man* (see Chapter 4.5).

The original scene graph only contains two plural labels: *people* and *men*. They are included in $\text{SG}_{original+attr}$ but they are removed before the plural generation to avoid redundancy and confusion. They are not as suitable to present object collections anyways because they do not necessarily cover all people or men in the image and lack a plural tag.

With the enhanced scene graph representation including plural objects, the phrase grounding accuracy improves from 47.93% to 49.63% using the vanilla aligner. On average, 17 plural objects are added per image, which is an increase of 28% in the average number of objects. Their large bounding boxes increase the chances that all objects mentioned in a plural phrase are covered and that the IoU score reaches 0.5.

An even larger performance boost comes with an aligner extension that differentiates between singular and plural phrases and pre-selects candidate objects based on number. This is implemented by parsing phrases with SpaCy [Honnibal and Montani, 2017] and checking for plural indicators. The phrase *two boys* for instance can be identified as a plural because of the *NNS* part-of-speech tag for *boys*. Besides part-of-speech, specific keywords also trigger the plural tag, such as *group* or *crowd* or *collection*. On the other hand, some *pluralia tantum* (nouns with no real singular form) are considered singular, for instance *jeans* or *pants*. They appear in plural form but they are usually represented by a single object. Interestingly, the scene graph represents *jeans*, *pants* and *shorts* with the singular labels *jean*, *pant* and *short*, although the box covers the entire piece of clothing. We changed the labels to plural form before embedding them as a word vector in order to avoid wrong associations. On the other hand, we do not alter the singular form label *glass* since it is ambiguous and represents both a vessel for drinking and a vision aid (*glasses*).

If the phrase to be aligned is a plural phrase, we discard all scene objects without a plural tag. Conversely, we only allow non-plural scene graph objects for singular phrases. 16% of all Flickr30k Entities are identified as plural phrases. For singular phrases, the pre-selection does not improve phrase grounding accuracy. In fact, when removing the plural objects as candidates, the accuracy slightly drops from 53.26% to 52.50%, probably because plural boxes often match with ground truth boxes for singular phrases. We have already discussed this problem in regards to the merging method in Chapter 3.3.1. For plural phrases, on the other hand, the performance increases from 36.63% to 56.46% when allowing only plural objects as candidates. The overall performance for both singular and plural phrases on $\text{SG}_{attr+plurals}$ is 53.36%, which constitutes a 4% increase compared to the vanilla aligner and a 5.5% increase compared to the original scene graph representation $\text{SG}_{original+attr}$. The plural objects extend the scene graph in a general way which is not only useful for phrase grounding but many other tasks. The differentiation between singular and plural phrases should be considered an essential part of unsupervised phrase grounding as only with this information reliable results can be achieved. Finally, in comparison to the uncontrolled box union approach as used in [Parcalabescu and Frank, 2020], our aligner generates fewer instances of boxes that are aligned correctly but have completely wrong labels (see an example in Figure 3.13).

The Flickr30k annotations for plural phrase entities either include one bounding box for phrases with collective terms (e.g. *crowd*) or several single bounding boxes for phrases describing multiple objects (e.g. *three women*). If there are several boxes, they are merged into one union box the same way we generate plural boxes. The coordinates of that union bounding box are used as the ground truth for evaluation. Unfortunately, the ground truth boxes for plural phrases often end up very large since they have to surround all single objects mentioned in the plural phrase – and they can be spread across the image, resulting in "empty" corners. Since our plural objects also exhibit this issue, using the IoU for evaluation is technically a fair way of computing whether they are a match. However, one should keep in mind that for large ground truth boxes many predicted boxes of medium or large size will be a match when evaluated with the IoU metric. The image size is limited so large boxes naturally overlap more often than smaller ones – hence, they are more likely to be considered correct predictions, regardless whether they really are. The accuracy of the baseline supports this claim: when simply predicting the entire image as the grounding area, the evaluation metric considers every fourth phrase as correctly grounded, even though the alignments have absolutely no value. This should be considered when evaluating enhancements that generate and promote

large bounding boxes, as not every improvement in accuracy necessarily implies higher quality alignments.

Rather than merging the boxes of the individual objects into a large union box when comparing ground truth and prediction for plural phrases, one could evaluate the individual objects separately. If three bounding boxes are assigned to the phrase *three children*, one could try to pair them with single scene graph objects (e.g. *child-1*, *child-2* and *child-3*). This enables an evaluation of which components of a plural phrase are found and which are not. It also eliminates the issue of "empty" corners and overly large bounding boxes. Future work should look into designing an advanced evaluation method that addresses these shortcomings and computes a less biased phrase grounding accuracy.

4.3 Object Reduction for Multiple Representations

We have discussed the issue of objects being represented by several partially overlapping bounding boxes, as well as the consequences of this over-representation in Chapter 3.2.3. In order to improve the quality of the scene graph and thereby the performance of phrase grounding, we attempt to automatically identify superfluous boxes and eliminate or merge them. As scene graph objects are removed, we generate a new scene graph representation ($SG_{attr+plurals+red}$), which contains fewer nodes than $SG_{attr+plurals}$ but more than $SG_{original+attr}$ since the plural objects are still included.

Figure 4.3 shows an example image that demonstrates the issue of multiple representation. The scene graph for this image contains 18 people objects, while in reality there are only 4 or 5 people visible (the man, the girl and 2 or 3 people in the background). Figure 4.4 shows the reduced version with accurate bounding boxes and suitable labels. Object reduction is especially challenging for people objects because they are often represented by several objects with various person labels. So besides identifying which boxes represent the same person, we also have to select the best label. Figure 4.3 demonstrates this issue: the man is represented by the labels *man-1*, *boy-1*, *girl-2* and *woman-1* (in fact, it is not quite clear whether *woman-1* and *girl-2* are meant to represent the girl or the man). While *man* is the best suited label, the box *man-1* does not entirely cover the man in the image because the right hand and legs are outside the box. In order to ideally represent the man, the bounding boxes for *man-1*, *boy-1*, *girl-2* and *woman-1* need to be merged into one big box that receives the label *man-1* (see Figure 4.4). The girl, on the other hand, is represented by the boxes *child-1* and *girl-1*, both of which cover the area

Figure 4.3: Original bounding boxes.

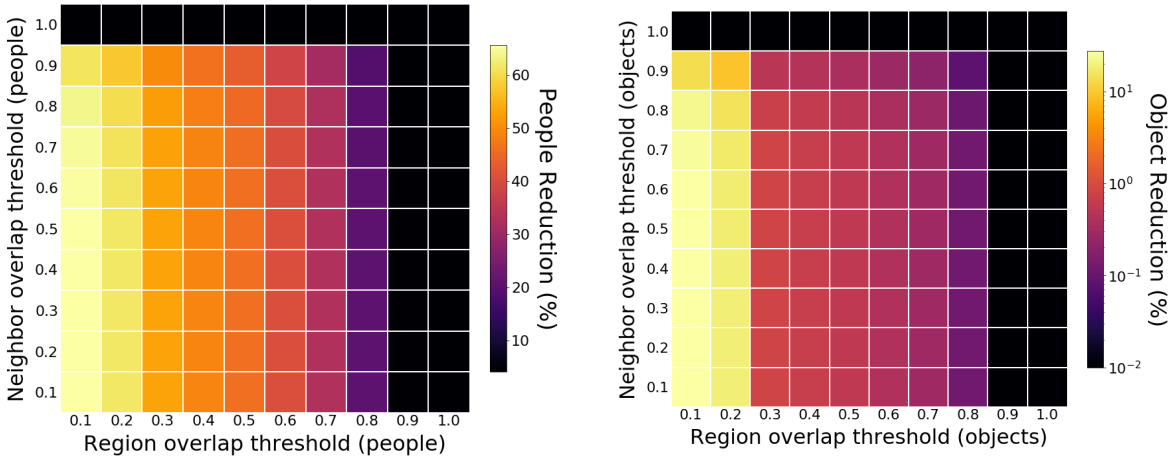
Figure 4.4: Reduced bounding boxes.

of the girl accurately and the labels are suitable. Still, the double representation is not ideal so the boxes have to be merged and the more precise label *girl* should be selected (see Figure 4.4).

The essential task for reducing multiple boxes is to identify which bounding boxes represent the same person or object, so we first have to examine which labels can be used to refer to the same object. People can be represented by any label in the people set. Multiple boxes for non-people objects usually have the same label (see Figure 3.8) but sometimes semantically similar labels are used. For this reason, the following labels are put together in sets: truck and car; motorcycle and bike; book and paper; glass and cup; table and desk and stand. In the next step, the objects for a given label set (e.g. all people, all surfboards or all cars/trucks) are divided into groups, each representing exactly one object from the image. For the image in Figure 4.3, one group would contain *child-1* and *girl-1* and another *man-1*, *boy-1*, *woman-1*, *girl-2*. There would also be two or three groups representing the people in the background. The grouping is achieved by comparing each pair of objects and computing two scores: the region overlap and the neighbor overlap. The region overlap is computed using the IoU of the object bounding boxes; it thus represents how much the two boxes overlap in the image. The neighbor overlap represents how many neighboring nodes the two objects have in common in the scene graph. It is computed by dividing the number of common neighbors by the total number of neighbors, so it can be understood as the graph-based counterpart of IoU. For people, only the neighboring nodes of the categories *body parts* and *clothes* are considered since two people objects are likely associated with the same body parts and clothes if they represent the same person. While the region overlap compares the size and position of the bounding boxes, the neighbor overlap compares their

contexts. If the region overlap and neighbor overlap exceed a certain threshold, the two objects are considered to represent the same object and are sorted into the same group. If an object from Group A and an object from Group B show high overlap, the two groups are merged into one. Finally, the bounding boxes for all objects in a group are merged to generate the final bounding box for that group. We opted for merging rather than selecting the largest box because the largest box is not necessarily precise (see *man-1* in Figure 4.3). Note that this may create boxes that are slightly too large. We select the label of the largest bounding box to represent the group since larger boxes tend to be labeled more accurately than smaller ones. (We also conducted experiments using the most general, most specific or most frequent label instead, as well as the label of the object with highest scene graph degree but none of them outperformed this approach.)

In order to determine the best *Region Overlap Threshold* (ROT) and *Neighbor Overlap Threshold* (NOT) we conduct parameter screening with a separate training set of 500 images. Screening requires a metric that represents how well the object reduction improves scene graph quality. However, automatically computing such a metric is not trivial as there is no complete set of gold annotations to compare the scene graph objects with. The Flickr30k Entities dataset only provides localizations for the objects mentioned in the captions – and without phrase grounding the scene graph objects are not linked to those localizations. Nevertheless, we attempt to exploit the Flickr30k annotations for computing evaluation metrics. For each Flickr30k entity in the training set, we check how many objects exist in the scene graph that could possibly be a match for the entity, so we get an approximate measure of whether the entity is over- or underrepresented. A scene graph object is considered a match for an entity if their IoU is greater than 0.5. For people entities, we check in addition whether any of the phrases with that entity ID show signs that the entity is female, male or a child, then discard all objects that do not fit. For instance, if an entity is tagged as *female* and *child*, we discard all objects that do not have the label *girl* or *child*. Ideally, there is exactly one scene graph object match for each entity. Excessive matches are counted as false positives, missing matches as false negatives. Note that this measure is obviously flawed: even the best scene graph representation would yield false positives, a) because the different entities often overlap so correct objects could have high IoU scores for several gold entities, and b) because the scene graph contains objects that are not represented in the Flickr30k entities as they are not mentioned in the captions. Nevertheless, this evaluation method allows us to compute precision, recall and F1 score, and while these scores should not be interpreted on their own, they allow comparisons between different parameter settings.



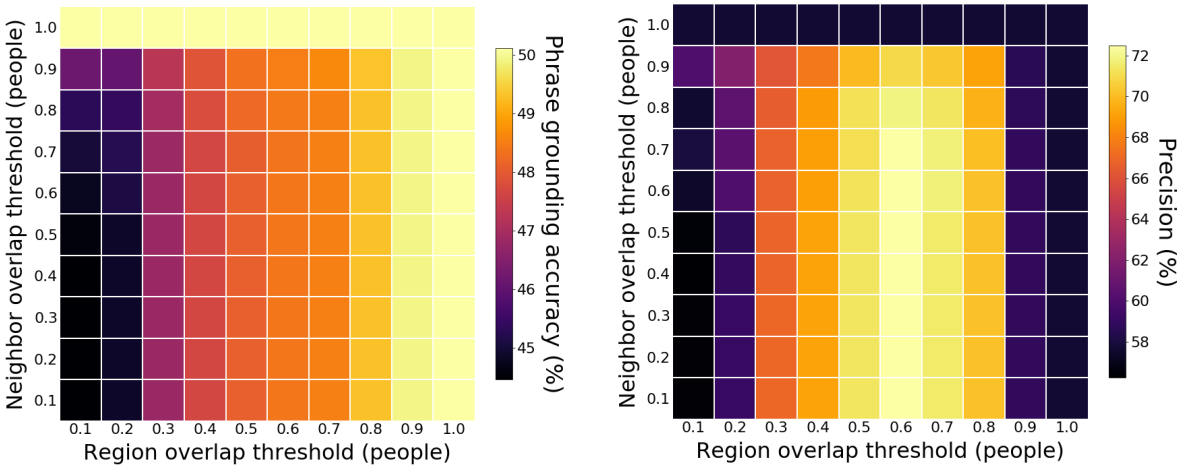
(a) Reduction rate (people).

(b) Reduction rate (objects).

Figure 4.5: Heatmaps for parameter screening (reduction rates).

As a final metric in the parameter screening, we use the phrase grounding performance. As mentioned previously, multiple boxes do not necessarily impair the phrase grounding system as it generally benefits from high recall. The removal of boxes and labels as well as the possibility of introducing errors in the process can hurt the system more than multiple boxes would. Therefore, the phrase grounding performance is generally not a good measure for evaluating object reduction. However, it is important to keep this metric in mind as we want to improve both the scene graph quality *and* the phrase grounding system, so we ideally do not want to enhance one at the cost of impairing the other. The heatmaps in Figures 4.5, 4.6 and 4.7 show the reduction rate, precision and phrase grounding accuracy as well as F1 scores for the different parameter settings for region and neighbor overlap thresholds (0.1 to 1.0 in steps of 0.1). We evaluate the people objects and the non-people objects separately since they are processed slightly differently and have individual thresholds.

Figure 4.5a shows how lowering the thresholds gradually reduces more and more people objects, from no reduction at the highest threshold level to a reduction of 60% of people objects at the lowest thresholds. Except for the columns with ROT <= 0.2, the reduction rate only visibly changes along the x-axis, which may mean that the neighbor overlap does not fluctuate as much. A similar effect can be observed in Figure 4.5b for the non-people objects, although on a smaller scale (note the logarithmic scale of the object reduction rate). The highest achieved reduction for non-people objects is merely 27% and when ROT > 0.2 the reduction rate drops below 1%. The reduction rate is so much lower for the non-people objects than for



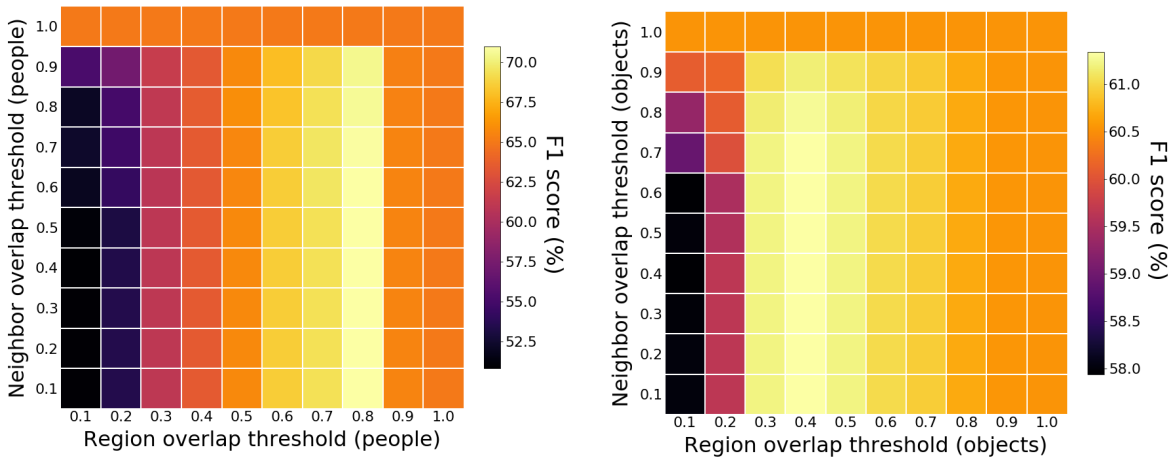
(a) Phrase grounding accuracy (people). (b) Precision (people).

Figure 4.6: Heatmaps for parameter screening (accuracy and precision).

people objects because non-people objects are six times as frequent. In addition, only objects with the same or a similar label can be merged, while people objects can be merged with any other people object.

Figure 4.6a shows the phrase grounding accuracy with people object reduction. It comes as no surprise that the accuracy improves with higher thresholds and lower reduction rate. In fact, the best performance for both people and non-people objects is achieved with no reduction at all. The same is true for the recall measure, which emphasizes that phrase grounding accuracy and recall highly correlate. The difference in phrase grounding accuracy between the lowest and highest reduction rate is 6%, while for non-people objects it is only 1% because the reduction rate is lower. As for the recall, with people object reduction the recall drops from 83% to 43% and with object reduction for other objects from 80% to 74%.

There are, however, metrics that increase with the reduction of superfluous objects. In contrast to recall, the precision increases by 16% as the ROT approaches 0.6 (40% reduction rate) for people objects. Figure 4.6b shows the highest precision is reached with ROT = 0.6 and NOT <= 0.7 and that with divergence from these thresholds (to either side) the precision gradually drops. This demonstrates that precision does not continuously increase with higher reduction rates; at some point the effect is reversed. For non-people objects, the highest precision increase is only 1% when reaching the ROT of 0.4 (0.7% reduction rate) but the overall effect is the same.



(a) F1 score (people).

(b) F1 score (objects).

Figure 4.7: Heatmaps for parameter screening (F1 score).

In summary, recall and phrase grounding accuracy are highest when not reducing the number of objects at all, while precision increases when approaching a region overlap threshold of 0.6 or 0.4, respectively. Therefore, the next step is to find a middle ground that allows object reduction to improve the scene graph quality and precision but does not drastically decrease the phrase grounding accuracy and recall. The F1 score, which combines precision and recall provides the optimal solution. Figure 4.7a shows that for people objects the ideal settings are a ROT of 0.8 and a NOT below 0.7; exactly in the middle of the settings for best recall and best precision. With these settings the F1 score is 77%, which is 12% higher compared to the version with no reductions. Precision is 70% (+13%), recall is 78% (-5%) and phrase grounding accuracy is 49% (-1%). For non-people objects the highest F1-score is found at a ROT of 0.4 and a NOT below 0.7 (see Figure 4.7b), same as for the highest precision. Precision and F1 score increase by 1% in comparison to the non-reduced version, recall and phrase grounding accuracy do not change. Since these thresholds provide balanced results for both scene graph quality and phrase grounding, we select them as final settings for the object reduction system: ROT = 0.6 and NOT = 0.3 for people objects, and ROT = 0.4 and NOT = 0.3 for non-people objects.

The heatmaps and scores show that the neighbor overlap threshold does not make a significant difference but we choose to keep it included since it may help in some cases and can be extended in future work. An inspection of people and their neighboring objects reveals that in cases where people are standing very close to each other (e.g. in a crowd), the body parts and clothes are not always connected to the correct

	Original	After reduction
Reduction rate	0.0	3.0
Precision	55.2	59.5
Recall	79.9	78.2
F1 score	61.6	63.6
Phrase Grounding Accuracy	50.0	49.1

Table 4.1: Phrase grounding accuracy before and after object reduction.

person. Although we do not assume that this is a serious source of errors, the neighbor overlap threshold could be adjusted to accommodate for such mistakes, for instance by lowering the threshold for images/areas with many people and raising it when people are depicted clearly separate. Another finding is that the reduction rate for non-people objects is so low that it hardly has any impact, although it may still improve the scene graph quality in ways not measured by these scores. As the non-people objects include a wide array of object types, future work could extend the merge criteria depending on the object category or type in order to increase and improve object reduction.

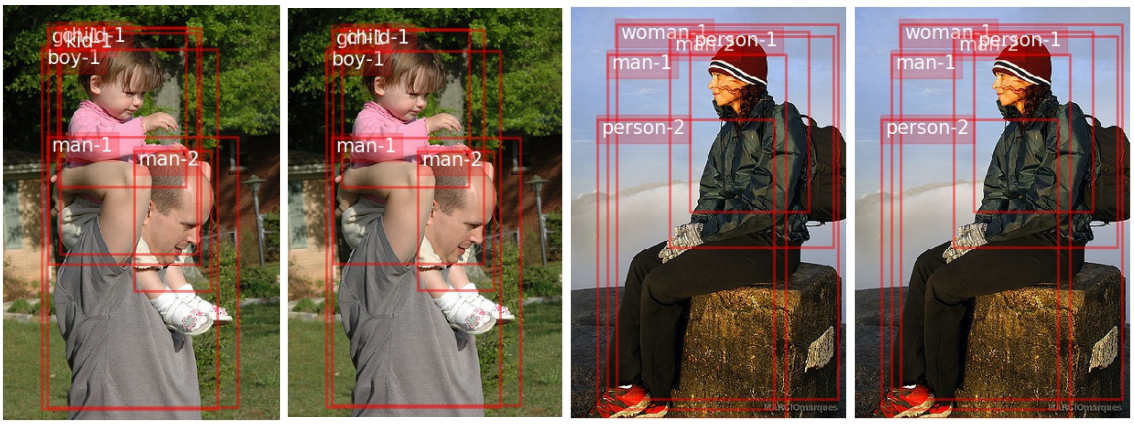
The scores for the complete evaluation on the training set with both people and non-people object reduction are summarized in Table 4.1. F1 score increases by 2% and precision increases by 4%, at the cost of recall and phrase grounding accuracy dropping by 1% each. The scene graph is trimmed by reducing the people objects by 20% and the non-people objects by 0.7%, giving an overall reduction rate of 3%. We consider the object reduction a success because the quality of the scene graph is improved while maintaining a similar phrase grounding accuracy. By reducing superfluous boxes, the scene graph becomes clearer and other scene graph extensions become more accurate, for instance the generation of plural objects. (Note that while the object reduction is described after the plural object generation in this thesis, object reduction is applied *before* plural object generation in the pipeline used for generating the representation $\text{SG}_{attr+plurals+red}$).

Finally, we evaluate the new scene graph representation $\text{SG}_{attr+plurals+red}$ by computing the phrase grounding accuracy on the test set. As expected, the accuracy drops by roughly 0.7% compared to $\text{SG}_{attr+plurals}$ for nearly every settings constellation. This confirms that phrase grounding and its evaluation metric are not sensitive to superfluous boxes and that a higher quality scene graph representation does not always pay off.

(a) Before reduction. (b) After reduction. (c) Before reduction. (d) After reduction.

Figure 4.8: Successful people object reductions.

For a qualitative evaluation, we manually inspect 50 images from the test set. We focus on people object reduction since its effect is more visible, so we discard images without people in it as well as images with more than 10 people (e.g. crowds), as the reduction cannot be studied well on those. For each image, we count the actual number of depicted people and the number of people objects in scene graph before and after reduction. On average, there are 2.6 people in an image (foreground and background combined). The original scene graph contains on average 7 people objects, the reduced version only 4.8, which corresponds to a reduction rate of 31%. The original scene graph contains on average 4.4 superfluous people objects, the reduced version only 2.2; the error is halved thanks to the reduction. There are only two cases in which two boxes are merged that are not supposed to be combined. The examples show that the reduction works best when the multiple boxes are of similar size and shape, as demonstrated by the skateboarder in Figures 4.8a and 4.8b. Object reduction is also more successful when the entire body is shown and the person is in a standing position (see Figure 4.8). People in sitting or squatting positions or people for which only the upper body is shown are generally handled poorly, both in the original and reduced version (see Figure 4.9). The woman in Figure 4.9c is represented by many differently sized and shaped bounding boxes, so reduction is not possible because the region overlap is not high enough for any box pair. Some people receive an additional bounding box that covers their head only, for example *man-2* in Figure 4.9a. Although *man-2* is entirely covered by the box for *man-1*, they are not merged since the *IoU* is too small. Such cases occur frequently and are usually not resolved correctly. However, simply merging or removing all people boxes that are covered by another would not be an ideal solution either, since they just as often represent two different people, for instance the girl



(a) Before reduction. (b) After reduction. (c) Before reduction. (d) After reduction.

Figure 4.9: Unsuccessful people object reductions.

and the man in Figure 4.4. One could attempt to separate the two cases in future work by merging all boxes that merely contain a head, but not the others. The label selection for the merged boxes produces good results in general. The people for which only one bounding box remains in the end usually receive a suitable label (see Figure 4.8). The few errors show that there is a tendency towards male labels and adult labels, so the most frequent incorrect label is *man*, as shown in Figure 4.11, in which the little boy is labeled *man-1*.

Interestingly, for blurry images or images with no people in it, random objects (backpacks, poles, chairs) or animals are detected as people, possibly because the object detector expects people in the image. Multiple boxes for these objects are usually not reduced, probably due to missing body parts and clothes that are needed to reach the neighbor overlap threshold. In future work one could try to eliminate these boxes entirely by comparing the neighboring nodes to real people's neighboring nodes.

In general, the object reduction improves the scene graph representation although not yet to a satisfactory level. The reduction methods prefers false negatives over false positives, as there are clearly more missed mergers than incorrect ones, probably due to the high thresholds set for maintaining phrase grounding accuracy. The analysis also showed that our reduction method does not cover some cases (for instance boxes completely covered by others) and is affected by incorrect people labels for non-people objects. Future work should look into more advanced and case-specific reduction criteria, possibly incorporating the neighboring nodes, perspective or posture. It should be noted that future object detectors will most likely become more precise in identifying people and aid in producing more accurate scene graphs that

may not include any multiple bounding boxes for the same object at all. Therefore, there may be no need in the medium to long term to further improve object reduction and achieve perfection. However, it is important to be aware of the multiple object issue and its consequences for scene graph quality, especially when extending the graph based on these original objects. Finally, the fact that the multiple objects do not significantly affect phrase grounding accuracy is a clear indicator that the phrase grounding evaluation method may not measure all relevant aspects, as it is indifferent to or even supports superfluous and incorrect bounding boxes.

4.4 Color Attribute

We discussed in Chapter 3.2.4 that color constitutes an important attribute, so we add this attribute to the scene graph and include it in the aligner process. Figure 4.10 shows the relevance of color for some alignments. The image shows two baseball players in differently colored shirts. The caption describes *shirt-1* and *shirt-2* as *green shirt* and *red shirt*, respectively. The label *shirt* is not sufficient to differentiate the two, so the color attribute is required. In order to compute the color for each object, we first crop the image to the area of the bounding box region and represent it as a numeric array (RGB triple for each pixel). We group the various colors present within a bounding box into separate major groups, from among which the most prominent is then selected. Thus, we employ k-means clustering to group the pixels into 5 different color clusters and then use a centroid histogram[2] to count how many pixels belong to each cluster. We consider the centroid of the most frequent cluster to be the most dominant color and use its RGB value as the representative color of that object. This method yields a more distinct approximation than simply taking the mean over all RGB values, which could be a color that itself does not appear at all within the image section. We add 20 to every color value in the final RGB triple in order to correct for the fact that images are usually perceived as brighter on screen than they are in reality since the light of the screen adds to brightness and affects perception, including that of the human Flickr30k annotators. Besides the RGB triple, we also retrieve an approximate color name selected from a set of 10 base colors. Thus, each scene object receives two color attributes, an RGB triple and a word.

In order to incorporate this new attribute into phrase grounding, the colors mentioned in phrases also need to be identified and brought into numeric form. Since we compute the color name for each scene graph object, it would also be possible to

[2] `https://gist.github.com/andrisgauracs/d876674a486f6c02123389e763f57dc3` (Last accessed: 21.09.2020)

Figure 4.10: Red shirt vs. green shirt.

Figure 4.11: Foreground vs. background.

directly compare the color name with the color specified in the phrase, for instance by measuring the distance in a word embedding space. However, for computing differences in color perception, numeric values in a color space are more reliable than word embeddings.

If the phrase contains a color found in the set of 148 colors from the python library *matplotlib*[3], the color name is transformed into its RGB triple. This allows us to compare the phrase color with every scene graph object color. Importantly, to measure the difference between two colors the CIELAB color space is more suitable than the RGB color space as it is perceptually uniform and the Euclidean distance between colors in CIELAB space correlates strongly with the color difference perception of humans, which is not the case for the RGB color space [Tkalcic and Tasic, 2003]. Thus, we transform the RGB triples for phrase and object colors to the CIELAB color space.

In the phrase grounding process, we first retrieve the label with the highest cosine similarity for the phrase as usual. If there are several scene graph objects with this label, we would normally select the largest bounding box as a default solution. However, with the added knowledge about the object and phrase colors, we can rank the candidates based on their color difference. If the highest ranking label is *shirt* and there are two shirts in the scene graph, we select the one with the smallest color difference to the color in the phrase. In the example of Figure 4.10, the Euclidean distance between the CIELAB value for the color *green* and the detected colors for *shirt-1* and *shirt-2* is 5389 and 7553, respectively. The distance is smaller for *shirt-1* so the phrase *green shirt* is aligned to *shirt-1*. The same applies for *red shirt* and *shirt-2*.

[3] `https://matplotlib.org/3.1.0/gallery/color/named_colors.html` (Last accessed: 21.09.2020)

If the shirts have similar colors, however, it is risky to diverge from the default solution (the largest box with that label), which is usually a good prediction. The color difference is rarely indicative in such cases; one object may simply be in the shade or include some other colors, which changes the overall color value. Therefore, we allow ranking by color only if there is a significant color difference between the highest ranking and the second highest ranking color object, specifically a Euclidean distance greater than 1500. In the example above, the Euclidean distances are 5389 and 7553, thus the difference is greater than 1500 and the alignment by color is allowed. (Note that our implementation relies on correctly predicted labels so if *shirt* is not the highest ranking label, the shirt objects will not be considered as candidates at all, despite their color similarity to the phrase.) The minimal distance constraint may be a simple implementation to avoid unnecessarily diverging from the default solution but in combination with the multiple representations issue it disables the color extension for many samples. If there are multiple boxes for the same object, they usually receive a similar color attribute because they are highly overlapping, so the color difference constraint does not allow alignment by color. In fact, the color difference is only high enough in 8% of all cases and the multiple boxes are partially responsible for that. Future work should look into adjusting the color distance constraint or finding another way to prevent aligning by color if the colors are very similar.

13% of all phrases contain a color. When evaluating only on those phrases, the color attribute achieves an improvement of 1% in accuracy, more precisely an increase from 51.70% to 52.8% on $\text{SG}_{original+attr}$. For the other two scene graph representations, the effect is reduced to 0.5%, possibly because the plural objects already boost the performance. Since there are relatively few color phrases and only for 8% of them the prediction is changed, the improvement on the entire dataset is merely 0.1% on all representations. Note that the color attribute only has a chance at improving if the highest ranking label provides a correct bounding box and if that bounding box is not the largest with that label (which would be selected by default). The color similarity does not (yet) allow switching to another label. We conducted an experiment in which we considered the top 25 scene graph objects as candidates, based on their label ranking. The top label usually does not provide 25 objects, so the next ranking labels fill up the set. From these 25 candidates, the one with the highest color similarity is selected, regardless of the label. However, this approach drastically reduces performance (by 10% when evaluating solely on color phrases), suggesting that color similarity alone does not provide reliable predictions, especially when the candidates have various labels. The color similarity helps the most for small candidate sets with the same label.

Despite the minimal improvement on the entire test set, the color enhancement should be considered a proof-of-concept, showing that attributes can in general aid phrase grounding. The addition of further attributes (including ones that affect more or different phrases), can provide valuable information for phrase grounding as well as other tasks.

The ranking by color difference (or color similarity, if reversed) could also be used in direct combination with the word embeddings cosine similarity measure. The scores could be normalized and combined, even weighted according to importance of the individual similarity measures. Since only few phrases contain colors and this approach does not enable the above mentioned minimum color difference constraint, we did not implement it here. When introducing other attributes, however, combining their (weighted) similarity measures may be a simple and elegant way for predicting the best candidate. Thus, future work should try to include more attributes that allow computation of similarity scores between phrase and scene graph objects, for instance size, pattern, texture, absolute position (e.g. *at the right corner of the picture*) or relative position (e.g. *next to the table*). Attributes complement the embedding-based cosine similarity as they can differentiate between objects of the same label.

4.5 Assistance for Grounding People Phrases

One third of the phrases describe people and two thirds of them are grounded correctly, as demonstrated by Figure 3.11. The incorrect alignments in this category are not caused by a lack of people objects (as shown by the high Flickr30k coverage) but by the diversity and inconsistent use of their labels. The people label set is large and includes hypernyms for other labels (e.g. *person* for *man*) as well as highly overlapping labels (e.g. *woman* and *lady*). The ambiguous use of terms such as *girl* adds yet another challenge: the scene graph uses the label *girl* for both a child and a young woman – and the captions show the same ambiguity. In addition to ambiguity, different levels of specificity are used both in phrases and scene graph labels. A little boy can be described or labeled very generally as a person or more specifically as a child, or even more specifically as a boy. If the caption phrase and scene graph label do not represent the same level (for example, the phrase *a little boy* and the object label *kid-1* as in Figure 4.12), aligning becomes more difficult. Word embeddings and their similarity scores are supposed to resolve such problems but they do not always succeed. For instance, *boy* is closer to *girl* (cosine similarity: 0.85) than to *kid* (0.63) so the phrase *a little boy* in Figure 4.12 would not be aligned

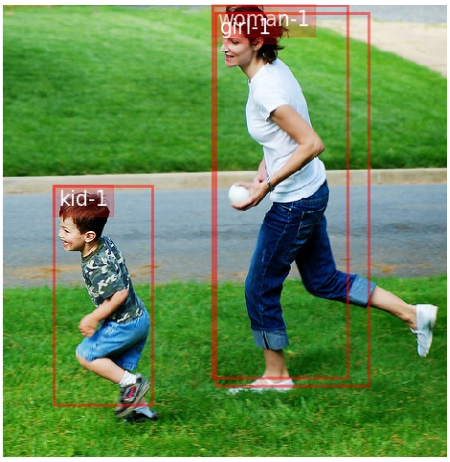


Figure 4.12: *kid-1* vs. *girl-1*.

Figure 4.13: Drums labeled as *basket*.

with *kid-1* but with *girl-1*, which is completely wrong. The same problem occurs with *woman* and *man*, which are closer to each other than to *person*. Of course, we cannot expect the cosine similarity to provide perfect rankings for all word pairs but since these are very basic and frequent phrases and labels, this shortcoming needs to be addressed. Parcalabescu and Frank [2020] include an additional similarity measure based on the shortest WordNet path to account for this issue, which may in fact correctly resolve this particular example as *boy* and *girl* are two steps apart in WordNet (connected through the hypernym *child*) while *boy* and *kid* are only one step apart. The multiple object representations for the same people add another layer of complication to the problem since one person can have several different labels (e.g. *girl-1* and *woman-1* in Figure 4.12).

In order to aid in the grounding of people phrases, we enhance the scene graph and the aligner with two extensions aimed at filtering out unsuitable scene graph objects – or at the very least decreasing their ranking among the candidates. We refer to them as *label preference* and *foreground preference*. The first approach pre-selects scene graph object candidates by searching for keywords in the phrase that give an indication about gender and age and then discarding all candidates with a label that is not found in the label set of this keyword. The keyword *girl* for instance allows the labels *girl*, *woman*, *lady*, *child* and *person*. This prevents the aligner from grounding any phrase containing *girl* with an object with the label *man*, *boy* or *guy*, unless there is no other candidate available. The keywords do not have to be single words, they can also be expressions, for example *little girl*, which in contrast to the keyword *girl* only contains *girl* and *child*, as it is unlikely to refer to a grown-up woman. We extract 66 terms and expressions describing people from 200 Flickr30k captions from the training set. The keyword list includes the scene graph people

labels plus words and expressions such as *baby, toddler, teen, youth, small girl, little boy, middle-aged woman, elderly man, daughter, father* and *someone*, many of them both in singular and plural form. The associated label sets are created manually to ensure high quality. Some keywords share the same label set, for instance *toddler* and *child*. The 66 keywords require 25 different candidate label sets.

In future work, the candidate label sets should be extracted automatically. For every phrase and scene graph object pair in the training set, one could compute the *IoU* and if it is greater than 0.5, the object label would be saved as a candidate label for this phrase. As key for saving the candidate labels, one could either use keywords extracted from the phrase or encode the entire phrases as word embeddings. In the latter case, all keys with high cosine similarity and high overlap in their label sets would be merged in the end. When grounding a phrase at testing time, the key with highest cosine similarity would be selected and its candidate label set would be used to filter for suitable candidates. This method could be extended to non-people objects and provide insight into how objects are labeled given the limited label set. For instance, *drums* are not in the label set but they are often detected and labeled as either *basket* or *umbrella* (see Figure 4.13). The label *basket* is used for single drums and *umbrella* for an entire drum set. Such connections could be learned automatically and incorporated into the alignment process, so that for the phrase *drum* objects labeled *basket* or *umbrella* are promoted, even if they are ranked low by the cosine similarity of word embeddings.

The second approach for improving the alignment of people phrases is to promote people objects in the foreground of the image. Captions usually focus on people in the foreground, often not even mentioning the people in the background, so the idea is to give higher preference to foreground objects when aligning people phrases. This helps in cases in which a background object has a label with higher cosine similarity to the phrase than the correct object in the foreground. An example is given by the boy in the foreground in Figure 4.11, who is incorrectly labeled as *man-1* (in this particular example as an effect of failed label selection during object reduction, but incorrect labels also occur in the original scene graph). In the background, there are two people labeled with multiple boxes (*person*, *man* and *boy*). The captions for this image describe the boy as *boy*, *child* or *kid*; the people in the background are only mentioned in two of the captions. The cosine similarity ranking suggests the label *boy* as the best candidate for the phrases describing the boy in the foreground – as it should. Unfortunately, the only *boy* object in the scene graph represents one of the people in the background (*boy-2*) and therefore the phrase is aligned to the wrong bounding box. This can be prevented by first looking for a reasonable candidate in the foreground and only considering background objects if no candidates are found.

The foreground preference is implemented as a scene graph attribute, consisting of a Boolean value that represents whether the people object represents a person in the background of the image. We decide whether a people object is in the background by comparing its size to the size of the largest people object. If the largest people object is more than five times larger than the object in question, it is considered to be part of the background and tagged as such. If the image shows no people in the foreground, the size of the largest object is usually so small that it does not trigger any background tags and all objects are considered equal candidates. When grounding a phrase, people objects with a background tag are only considered if nothing suitable is found in the foreground people set. As an extension, one could search the caption for prepositional phrases such as *(a woman) in the background* and lift this constraint for the phrase it is dependent on.

The label preference and foreground preference extensions are only applied to people objects and people phrases. The foreground preference is only used on singular people phrases. We use the keywords from the label preference to determine whether a phrase describes people or not, as we do not want to be dependent on the annotated phrase categories. With this method, we identify 27% of the phrases as people phrases, although we know from the Flickr30k analysis that it should be 34%. Thus, the keywords do not cover all possible ways to describe a person, for instance roles or professions. The missing 7% do not benefit from the grounding assistance extensions. To facilitate evaluating the two extensions, we test them separately on the people phrases only. The label preference improves the accuracy by roughly 0.4% and the foreground preference by 0.5%. Combined they raise the score from 69.13% to 69.99%. The label preference performs generally slightly worse than the foreground preference on all three scene graph representations, even minimally decreasing the accuracy on $\mathrm{SG}_{attr+plurals+red}$, probably because the candidate set is already reduced in that representation. In general, the effect of the two extensions is surprisingly small.

A manual inspection of some people samples shows that the extensions do in fact help in some cases but that they lead away from the correct prediction in others. The phrase *the man's car*, for example, would be correctly aligned to the object *car* but since the keyword *man* is found in it, the label preference suggests the object *man* as the only candidate. A similar problem occurs for the phrase *baby blue bottoms*, which is aligned to *child* rather than *pants*. This indicates that one should limit keyword search to the head of the phrase. We also find cases in which the correct object was incorrectly marked as background, which excludes it from the candidate set. In some cases this is caused by the fact that both adults and children are shown in the image. The boxes for adults are naturally much larger than those for children,

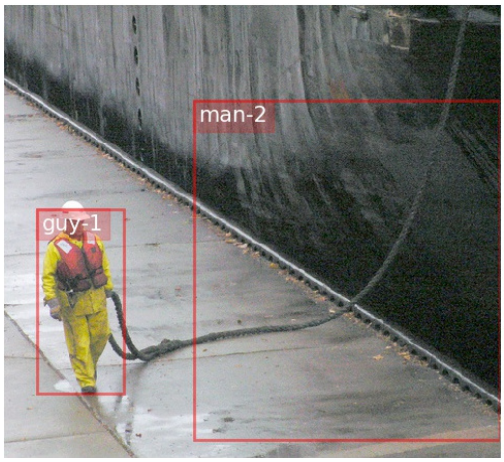


Figure 4.14: Incorrect person object.

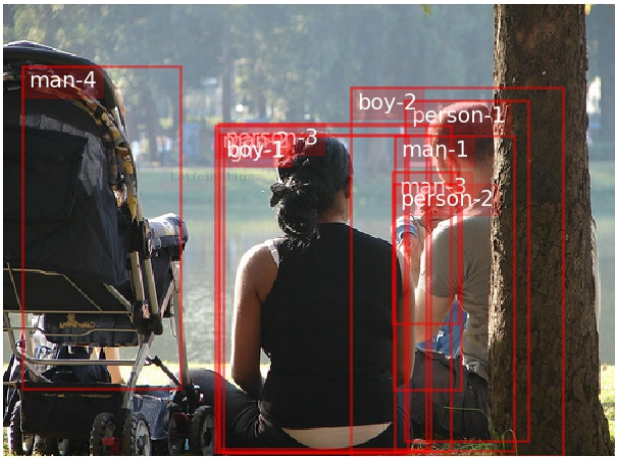


Figure 4.15: Numerous people objects.

so the children are erroneously labeled as background objects. This problem could be addressed by not labeling objects with the labels *child* or *kid* as background (at least if adults are shown in the image) but the problem would persist for *girl* and *boy* as they are ambiguous. However, the most common reason why people objects are incorrectly tagged as background are large people objects that do not really represent a person. Figure 4.14 gives an example with *man-2*. The object *guy-1* is labeled as background because it is five times smaller than *man-2* and so the phrase *worker* is aligned to *man-2*.

There are also cases for which the label and foreground preference may not lead to the correct bounding box but when evaluating manually we can observe a correction into the right direction. Figure 4.15 gives an example for an image with many people objects, highly overlapping and with unsuitable labels, which is not a rare case even after object reduction. The phrase *male* is originally aligned to *woman-1* since, strangely, the cosine similarity is higher for *woman* than for *man*. After the label preference extension the alignment is corrected to *man-1*. Unfortunately, the box for *man-1* is not large enough to count as a match. The boxes for *person-1* or *boy-2* would have been considered correct, although their labels are not ideal. This shows again that a correct label prediction does not guarantee a correct alignment, especially with multiple representations for the same person. For the phrases *infant* or *baby* (describing the baby the man is holding), the predicted label is *boy-1*, with and without label preference. The predicted box *boy-1*, however, represents the woman and not the baby. The box *man-3* would be the correct prediction but *man* is naturally not a label that is suggested for the phrases *baby* or *infant*. A correct alignment is nearly impossible at such a high level of imprecision among labels. Only the phrase *woman* experiences an actual improvement. Originally aligned to *man-1*,

it is now grounded with *person-3* (behind *boy-1*) because *man* is eliminated as a candidate. This image clearly represents the challenges multiple representations and wrong labels introduce for phrase grounding. The box *man-4* also provides another example for a random object identified as a person, which can confound the aligner. While label and foreground preference may improve the alignment in some cases, it still fails on images for which the people objects are convoluted and messy. It is reassuring, however, that they sometimes push the aligner into the right direction, even if it does not yet result in a successful alignment.

Evaluated on the entire test set, the people assistance extensions increase the aligner performance by 0.2% for $\text{SG}_{original+attr}$, 0.05% for $\text{SG}_{attr+plurals}$ and by 0.14% for $\text{SG}_{attr+plurals+red}$, only the first of which is statistically significant (p=0.049). The foreground extension performs equally or better on its own than in combination with the label preference extension, so the label preference method should be further refined. Automatically extracting candidate label sets for more keywords will most likely improve performance, especially when also applied to non-people objects.

Even if accuracy is not significantly improved, the people assistance extensions show that there are many ways to enhance the aligner and that simple attributes such as the background tag can aid in the alignment process. The analysis of the failed examples reveals the most prevalent challenges, for instance the high cosine similarities for words such as *girl* and *boy*, various ways of describing people (e.g. *teacher, daughter, worker*), including ambiguous language (e.g. *girl* for children and adults), and – yet again – the multiple representation issue and the evaluation metric, which favors correct boxes over correct labels.

4.6 Results and Discussion

In the subchapters above we explained how we generated three different scene graph representations and implemented three aligner extensions. We also discussed the individual results. Here we draw an overall conclusion and inspect how the different enhancements interact with each other. Table 4.2 summarizes all results. The asterisks (*) indicate a statistically significant improvement from the vanilla version of $\text{SG}_{original+attribute}$, which is 47.93%. One asterisk indicates $p <= 0.05$, two indicate $p <= 0.01$.

The addition of plurals objects to create the representation $\text{SG}_{attr+plurals}$ and its associated aligner extension for distinguishing singular and plural phrases constitute the most the successful enhancements by far. The enhanced representation alone increases the accuracy by nearly 2% and the number distinction adds another 4%.

Version	Accuracy		
	$SG_{original+attr}$	$SG_{attr+plurals}$	$SG_{attr+plurals+red}$
Baseline	23.63	–	–
Vanilla (before typo correction)	47.88	–	–
Vanilla	**47.93**	49.63**	48.93**
Number distinction	–	53.36**	52.58**
Color attribute	48.10	49.71**	49.02**
People assistance	48.13*	49.68**	49.07**
∘ Label preference	48.02	49.66**	48.87**
∘ Foreground preference	48.11	49.72**	49.22**
Color + people	48.30*	49.77**	49.15**
Number + color	–	53.51**	52.73**
Number + people	–	53.58**	52.86**
Number + color + people	–	**53.74****	53.01**

Table 4.2: Phrase grounding performance for different scene graph representations and aligner extensions.

The representation $SG_{attr+plurals}$ shows the highest scores for all versions and the number distinction is clearly the strongest extension. These enhancements not only increase accuracy, they also improve the quality of the scene graph and reduce the number of instances in which the correct box is predicted by chance.

The representation $SG_{attr+plurals+red}$ is meant tackle the multiple representation issue. While manual evaluation shows that the number of superfluous boxes is halved, it does not solve the problem entirely. Furthermore, it comes at the cost of losing nearly 1% in accuracy on the phrase grounding task across all versions since the reduced recall impacts performance. Multiple boxes affect all aligner extensions: they make it harder to group single objects into plural objects, they often disable the color extension because of the minimum distance constraint, and they interfere with the extensions for grounding people correctly. Therefore, solving this problem – either through better object reduction or a better object detector – is crucial. Even if the reduced representation does not yield a direct improvement on the vanilla version, it will boost the extensions. Finally, this issue reveals that the evaluation metric is indifferent to superfluous boxes and incorrect labels, revealing the need for new evaluation metrics that take this into account.

The color attribute and extension do not improve performance significantly, as they affect too few phrases and change the predicted box only for 1% of all phrases. Nevertheless, they show that the addition of attributes can improve the aligner in some cases, especially since only attributes can distinguish between objects of the same

label. This suggests that the inclusion of further attributes will most likely improve the aligner, while also adding to the information represented in the scene graph. Furthermore, the color extension performs well in combination with other extensions. The same increase in accuracy it achieves on the vanilla version is reached when combined with the people assistance and number distinction extensions. Thus, the color attribute provides relevant information the other extensions do not.

As for the extensions to assist grounding people phrases, the foreground preference provides better results than the label preference. This shows that a simple implementation in form of a scene graph attribute can have a greater effect than a complex system for selecting the best candidate label. Future work should consider that comparably simple solutions such as the number distinction and the foreground preference turned out to be more practical and efficient than complicated enhancements. As with the color extension, the people grounding assistance extensions work well – or even better - in combination with other extensions. They boost the color and number extensions by 0.2% on average. Especially the number extension benefits from them, possibly because they increase the accuracy for singular phrases, which are grounded slightly worse when plural objects are eliminated as candidates.

In general, aligner performance gradually improves as more extensions are added, which is a sign that the extensions address different aspects of the phrase grounding task and bring new and mostly non-conflicting information into the process. We achieve the highest phrase grounding accuracy on the representation $\mathrm{SG}_{attr+plurals}$ with all aligner extensions activated. This enhanced version with an accuracy of 53.74% clearly outperforms the original version of 47.93%. We gain an increase of 5.81%, of which 5.43% are due to the plural objects and the number distinction, 0.16% are caused by the color attribute and the remaining 0.23% are achieved by the extension for grounding people phrases.

The scene graph and aligner enhancements show that phrase grounding requires information on many different aspects of image and caption and that incorporating this information into the alignment process can be challenging. Nevertheless, our enhancement efforts generally improve the scene graph quality and the aligner performance. We improve the original system by 5.81% in accuracy. As the combination of various aligner extensions works well and increases the phrase grounding accuracy, we recommend adding further extensions in the future. The representation of the scene graph should be extended with further attributes and the multiple representation issue should be resolved. Finally, a new evaluation metric should be developed, attaching less importance to the predicted box and more to the label in order to exclude – or at least measure the impact of – correctly predicted boxes with an unsuitable label.

5 ConceptNet Extensions

Chapter 4 shows how we enhanced the scene graph with information that could be extracted from the image and captions. No external knowledge was added in the process. The enhancements focus on a clean representation of the scene graph as well as candidate filtering. Some scene graph labels are slightly altered by the introduction of plural objects and object reduction, but in general they are not changed. From the phrases we extract information about gender and age of people and whether they represents several objects or a single one, but the phrases are not altered in any way. While the extensions in Chapter 4 increase the performance of phrase grounding, they leave some alignment challenges unsolved, for instance specific vocabulary (e.g. *terrier mix*), abstract phrases (e.g. *concert*) or phrases describing roles rather than people (e.g. *construction worker*) as well as imprecisely labeled scene graph objects (e.g. *pole* for *hockey stick*). Thus, in this chapter we turn our attention away from improving the scene graph representation and focus on finding a bridge between phrase and scene graph label using ConceptNet.

ConceptNet [Speer and Havasi, 2012; Speer et al., 2017] is a knowledge graph that contains words and phrases which are connected with each other by edges representing their relation. The relations can be of hierachical nature (e.g. *IsA* or *PartOf*), they can represent which actions the concept is related to (e.g. *CapableOf*, *HasSubevent*, *UsedFor*, *ReceivesAction*), describe properties (e.g. *LocatedAt*, *MadeOf*, *CreatedBy*) or simply indicate a connection (e.g. *IsRelatedTo*). The relations can be symmetric (*IsRelatedTo*) or asymmetric (*UsedFor*, *IsA*). ConceptNet includes words from 304 different languages, 10 of which are considered the core languages with a large vocabulary and many connections. In total, ConceptNet comprises more than 8 million concepts and 21 million edges. There are 34 different relation types. For English alone, there are 1.5 million concepts. ConceptNet 5.5 is built from several sources and knowledge bases, for instance Open Mind Common Sense (OMCS), Wiktionary, WordNet and DBPedia. A triple including a concept, a relation, and another concept is called an assertion, for instance PAN $\rightarrow$ *UsedFor* $\rightarrow$ COOKING, CAT $\rightarrow$ *HasA* $\rightarrow$ TAIL, or PIANO $\leftarrow$ *IsSimilarTo* $\rightarrow$ CEMBALO. ConceptNet represents general knowledge in a graph structure and is incorporated in numerous applications to improve natural language understanding.

We have already mentioned ConceptNet in Chapter 3.2.4 and elaborated why it is not suitable for enhancing a static representation of the scene graph. The main reasons are the missing word sense disambiguation which can introduce noise, and the diversity and distribution of the relations which make it difficult to extract new concepts on the basis of relations, as some relations yield no results for some concepts and an abundance of results for others (which then have to be filtered). In other words, since the ConceptNet graph is so large and interconnected, it is difficult to decide in which direction or along which paths to search for useful information to be added to the scene graph. When expanding from one starting node, there is an overwhelmingly large number of possible paths to pursue. Therefore, we decided to incorporate ConceptNet not as a extension of the scene graph but as a link between scene graph and phrase, exploiting the fact that this allows us to look for a connection from both sides. If there are two nodes of interest, we gain information about direction and can limit our search to a subgraph of ConceptNet, which makes the process managable and the results more accurate.

Nevertheless, it is important to point out that even when working on subgraphs, ConceptNet comes with some caveats. First of all, ConceptNet is less curated than WordNet, has no hierarchical structure and is generally messier. ConceptNet allows edges between concepts that are not strongly related, for instance DOG $\leftarrow$ *Antonym* $\rightarrow$ COMPUTER or OFFICE $\leftarrow$ *Synonym* $\rightarrow$ POWER. Futhermore, assertions are sometimes not generally true, for instance DOG $\rightarrow$ *HasProperty* $\rightarrow$ BLACK. Also, not all concepts of the same type share the same types of edges. While *dog* has 12 properties (i.e. edges with the relation *HasProperty*), the concept *mouse* has none at all. This makes it harder to find general rules. Finally, the missing word sense disambiguation presents another issue. The ambigous concept *match* is related to *tennis match* as well as to *counterpart* and *combustion device*. The different meanings and their connecting concepts are not separated. Subgraph generation reduces the effect of some of these problems but they usually still persist.

In the subchapters below, we discuss how to generate a ConceptNet subgraph for each caption, how to extract support concepts for both scene graph labels and caption phrases and integrate them into the aligner, and how to employ similarity measures derived from subgraph graph metrics for phrase grounding. We evaluate the results and discuss our findings at the end of each subchapter and draw a general conclusion in the final one.

5.1 Subgraph Generation

ConceptNet is very large and working on the entire graph would be computationally taxing, so it makes sense to reduce ConceptNet to a subset of the graph including only nodes that are of interest. This also prevents to some extent unrelated concepts from interfering. Since we aim at connecting caption phrases with scene graph labels, we focus on their respective concepts in ConceptNet, the nodes on the paths connecting them, and their neighbors. We adapt the procedure of subgraph generation from Paul and Frank [2019] who use ConceptNet paths for predicting human needs to our task, and generate an individual subgraph for each caption/image pair.

The subgraphs are generated using source and target nodes as a starting point. Here, we define concepts extracted from the captions as source nodes and concepts derived from the scene graph as target nodes. We chose this terminology because the direction of phrase grounding is from phrase to scene graph object, although for the generation of the subgraph this directionality is not relevant. When extracting source concepts from captions (also referred to as *caption concepts*), we consider any n-gram up to four words with a noun in it as a candidate concept. We decided on this method because a noun is usually the essential part of a phrase but by merely taking the head of a phrase we would discard useful information. The n-gram approach enables the inclusion of compounds (e.g. fastfood restaurant), adjectives (e.g. young child) and verbs (e.g. sitting at table), which provides more diverse and precise concepts. For the caption below, the identified nouns are *child, table, fastfood, restaurant, piece, paper* and *pen*.

[EN#30976 A young **child**] sitting at [EN#30981 **table**] inside a **fastfood restaurant** and writing on [EN#30982 a **piece** of **paper**] with [EN#30977 a **pen**] .

For the noun *child*, we consider the following n-grams: *child, young child, child sitting, a young child, young child sitting, child sitting at, a young child sitting, young child sitting at*, and *child sitting at table*. A concept candidate may include several nouns as long as they are associated with the same entity ID (or have no ID at all). The nouns *child* and *table* are associated with two different entities (#30976 and #30981, respectively) so we discard the n-gram *child sitting at table*. The n-gram *piece of paper*, which also includes two nouns, is kept because they are part of the same entity. We apply this filtering to avoid having concepts representing two different phrases, which is problematic for phrase grounding. We keep nouns that are not part of any phrase (e.g. restaurant) because they provide context for the other concepts, even if they do not need to be aligned in the final task.

The n-gram approach splits the caption up in chunks, sometimes resulting in less valuable segments such as *of paper with*. This does not present a problem since we check for all candidate concepts whether there is in fact an entry in ConceptNet. This eliminates many unreasonable candidates. The look-ups are performed with both the surface form and the lemmatized form of the candidate concepts. From the 8 *child* n-grams above, *child* and *young child* are the only candidates that exist in ConceptNet; the rest is discarded. For the candidate concepts with a ConceptNet entry, we save the concept as well as the entity ID it is associated with, if there is one.

Extracting the target concepts (also referred to as *scene graph concepts*) is comparably easy. The scene graph object labels consist of one word each and all of them have a CocneptNet entry. Thus, the target concepts for a caption/image pair simply consist of the set of scene graph labels for that specific image.

Before generating the subgraph, we preprocess ConceptNet in order to remove generally irrelevant assertions and concepts. First of all, we discard all concepts that are not tagged as part of the English vocabulary. Although scene graph labels and (the head of) phrases are usually nouns, we refrained from only considering nouns since there may be valuable bridging concepts with other part-of-speech tags. However, we filter out rare vocabulary since an inspection of ConceptNet showed that there are many obscure concepts connecting frequent scene graph labels or phrases, for instance WOMAN $\leftarrow$ *RelatedTo* $\rightarrow$ MYRIOLOGUE or GIRL $\leftarrow$ *RelatedTo* $\rightarrow$ UNYAGO $\leftarrow$ *RelatedTo* $\rightarrow$ DANCE. Concepts such as *myriologue* (funeral song) and *unyago* (Swahili coming of age ritual) are very specific and link to other rare concepts (e.g. *extemporaneous*). They do not contribute to a general and meaningful representation of *woman* or *girl* and fail to provide useful links between concepts. For instance, *girl* and *dance* are generally better connected by the concept *ballerina* than *unyago*. Besides, scene graph labels and phrases usually contain relatively common words so there is no need to include rare vocabulary. We employ the *wordfreq* python library [Speer et al., 2018] which provides word and Zipf frequencies for many languages. The Zipf frequency projects the word frequency on a logarithmic scale (base-10 logarithm of occurrences per billion words), so a word with Zipf frequency 6 appears once every 1000 words and a word with Zipf frequency 5 only once every 10000 words. We discard all concepts with a Zipf frequency below 3. From the original 3423k English assertion triples, we discard 52% resulting in a reduced ConceptNet with 445k nodes and 1621k edges.

We transform the ConceptNet graph into an undirected graph since the directed graph would yield fewer paths between two concepts, as only paths with edges directed towards the target concept are returned when the subgraph is instantiated.

The path PAN → *LocatedAt* → KITCHEN ← *PartOf* ← STOVE would thus not be considered, despite connecting the concepts *pan* and *stove*. Since directionality does not matter for subgraph generation, we can work with an undirected version of the graph to avoid this problem. We do, however, save the direction of each assertion separately for later use. Finally, we discard all concept nodes that only have one neighbor or none at all (degree <= 1) since those concepts usually represent infrequent words and do not contribute much to the graph as they are not well-connected. This reduces the graph further to 183k nodes.

At this point, we have created a reduced ConceptNet graph with inter-connected nodes representing common English vocabulary. The next step is to generate a subgraph for each caption/image pair based on the extracted source and target concepts. The subgraph is generated by identifying relevant concepts and removing all other concepts from the reduced ConceptNet graph. We select the relevant concepts in three steps. First, we add all source concepts extracted from the caption and all target concepts extracted from the scene graph to the subgraph concepts list. Second, we compute all shortest paths between each source and target pair, as well as all shortest paths connecting two source concepts. We do not consider shortest paths between the target concepts (scene graph objects) since there is not much to gain from these connections. Furthermore, scene graph labels are not always correct and we do not want to propagate those errors. We could now simply use the concept nodes from all shortest paths but not all paths and nodes are expected to be relevant, so futher subselection is required. Paul and Frank [2019] include graph metrics such as closeness centrality, PageRank, and personalized PageRank for selecting relevant nodes and paths. We incorporate only the personalized PageRank.

The *PageRank* represents the importance of a single node within a graph, based on the incoming nodes and their importance. It is computed by counting how often each node is visited in a random walk. A random walk can either follow outgoing edges or jump to a random node with a certain probability. The *personalized PageRank* is a variation which represents the importance of nodes in regards to a specific topic (i.e. a set of starting nodes). Jumps in the random walk have to lead back to one of the starting nodes, meaning that personalized PageRank is biased towards the starting nodes and more localized than the PageRank. We use the personalized PageRank to determine which target concepts (scene graph concepts) are most relevant for which source concepts (caption concepts). For each source concept, we identify its 5 most relevant target concepts and add the nodes from the shortest paths connecting the source concept with any of the 5 target concepts to the subgraph nodes. The same procedure is applied to paths among source concepts. This selection further refines the subgraph and reduces its size. Without personalized PageRank selection, the

average subgraph would contain 74k nodes. With the selection, the average subgraph size is reduced to an average of 203 nodes.

The shortest paths in combination with the personalized PageRank partially resolve the issue of word sense disambiguation as they promote concepts and paths that are strongly related to the caption phrases. The concept *bat*, for instance, is ambiguous as it can represent both an animal or sports equipment. This could introduce unrelated concepts into the subgraph. However, if a caption concept contains *baseball*, only the sports-related neighbors of *bat* will be extracted, so for instance *strike* or *pitch* but not *flying mammal* or *sonar*. The personalized PageRank further promotes paths between relevant concepts and caption concepts.

Finally, as a third step we add all direct neighbors for each extracted concept to the subgraph nodes, which increases it to an average of 35k nodes. Future work could experiment with only including a subset of the direct neighbors as this step adds a very large number of concepts and reinforces the word sense disambiguation issue. The final subgraph for a caption/image pair consists of all scene graph concepts and phrase concepts, relevant paths among them and their immediate neighbors. Note that subgraph generation is computationally expensive, especially because of the shortest path computation for a large number of concept pairs. We limit the maximum path length to 10, as we are not interested in connections that are further apart. This limit also reduces processing time as the search space is reduced. Still, subgraph generation is time-consuming, so in order to retrieve a reasonable number of samples we compute them in parallel by multi-processing.

Figure 5.1 shows an example for a reduced subgraph, built from the caption concepts and scene graph concepts extracted from the image and caption in Figure 5.2. It does not represent the complete subgraph as it would contain too many nodes to display in a readable fashion. Using the generated ConceptNet subgraph, we develop two different approaches for improving phrase grounding. The first approach extends the semantic field of scene graph labels and phrases by incorporating support concepts. The second approach uses graph metrics such as personalized PageRank and shortest path length from the subgraph to measure the connectedness between phrase concept and candidate scene graph concepts. The first approach enhances the established aligner used in Chapter 4, while the second approach develops similarity measures for phrase grounding that are independent of this pre-existing system.

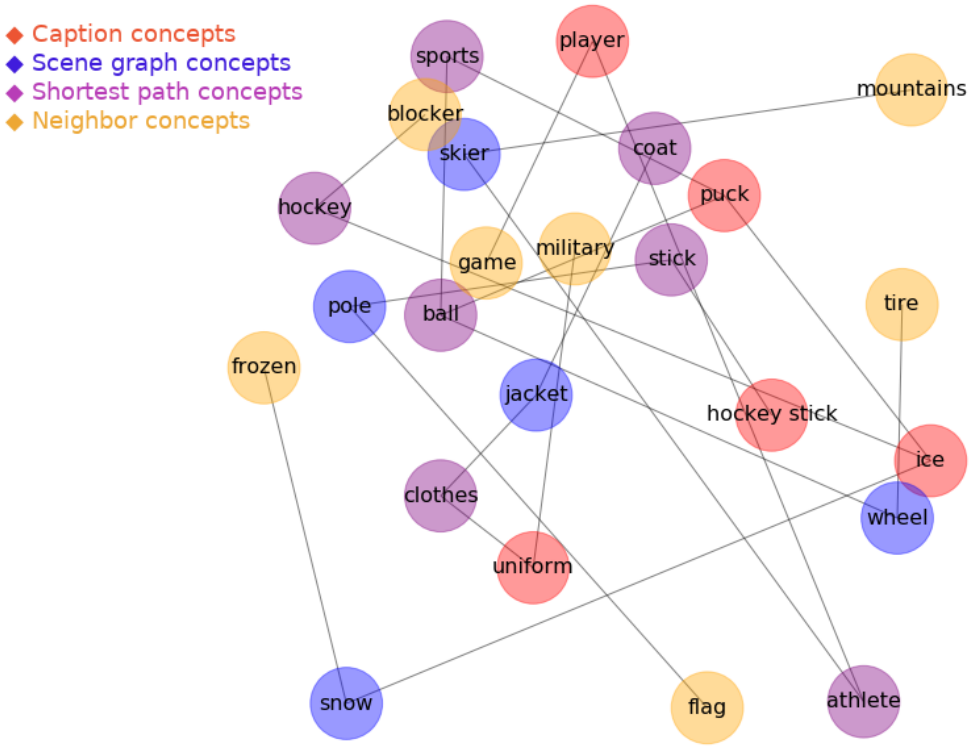


Figure 5.1: Reduced subgraph example for image and caption in Figure 5.2.

5.2 Support Concepts

Phrases often contain specific terminology (e.g. *terrier mix*) and abstractions such as scenes (e.g. *concert*) or roles (e.g. *musician* or *tennis player*) that cannot easily be aligned to the scene graph objects. The scene graph labels constitute common words and due to their limited number they are not precise (e.g. *pole* for *hockey stick* in Figure 5.2). By incorporating world knowledge extracted from ConceptNet, we attempt to overcome these challenges. We find support concepts for both phrase concepts and scene graph concepts that are supposed to expand the semantic space and improve phrase grounding. For instance, the assertion TERRIER $\rightarrow$ *IsA* $\rightarrow$ DOG will help to align the phrase *terrier mix* to *dog*. The assertions POLE $\leftarrow$ *IsRelatedTo* $\rightarrow$ STICK $\leftarrow$ *IsA* $\leftarrow$ HOCKEY STICK presents an aid for aligning the phrase *hockey stick* to *pole*. We include the support concepts by computing the mean of the embeddings vector of both the original concept and the support concept, which is then used to compute the cosine similarity to the phrase.

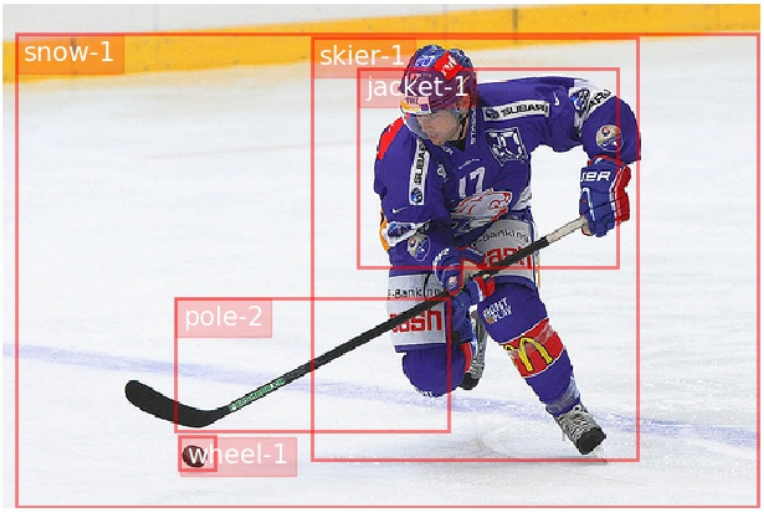


Figure 5.2: *A male hockey player in a bright purple uniform skates across the ice with a hockey stick and a puck.*

5.2.1 Collecting Candidates

The search for support concepts is conducted on the subgraph within a 3-step range from the scene graph concept or phrase concept in question. Even on the reduced subgraph, there is usually a large number of candidate concepts, so a general method is needed for selecting the best. The examples above suggest that the relations between the concepts could be an indicator for how suitable the concept is for improving the aligner. In those examples, the relation *IsA* and the relation tuple *IsRelatedTo*, *IsA* lead to concepts that help the aligner. Unfortunately, this is not always the case. Applied to the concept *bag*, for instance, the path BAG $\leftarrow$ *IsRelatedTo* $\rightarrow$ PROPERTY $\leftarrow$ *IsA* $\leftarrow$ CONSISTENCY leads to a concept that will most likely not support the scene graph object *bag*, for example when aligning it with the phrase *purse*. Other relations such as *Antonym* may never return a useful support concept at all (e.g. SNOW $\leftarrow$ *Antonym* $\rightarrow$ SUMMER). This shows that relations used for extracting support concepts need to be carefully selected. Since there are 34 different relations, which could make up 40k different relation tuples on a 3-step search space, manual evaluation is clearly not an option. Therefore, we implement a simple learning algorithm for ranking the relations based on their success rates, which then allows the best support concepts to be selected for any concept node. We learn these success rates separately for scene graph labels and phrases as they may benefit from different relation tuples.

	hockey player	uniform	ice	hockey stick	puck
skier	**0.28**	0.01	0.31	0.03	0.17
jacket	0.09	**0.44**	0.14	0.11	0.08
snow	0.02	0.06	**0.53**	0.11	**0.26**
pole	0.08	0.02	0.19	0.12	0.13
wheel	0.08	0.04	0.13	**0.14**	0.19

Table 5.1: Original cosine similarity adjacency matrix.

As a training set, we use 500 images with 2500 captions and 85k phrases. The subgraphs are computed for each caption and image pair and candidate support concepts are extracted within a 3-step range. We save the relation tuples leading to those concepts as well as the edge direction, for instance *IsRelatedTo* $\leftrightarrow$, *IsA* $\leftarrow$. Note that the subgraph is undirected but we have saved the direction for all edges beforehand, so they can easily be retrieved. The direction is relevant for asymmetric relations such as *IsA*. Consider the scene graph label *dog* as an example. While the assertion DOG $\rightarrow$ *IsA* $\rightarrow$ ANIMAL provides a hypernym for dog, an assertion with the opposite direction such as DOG $\leftarrow$ *IsA* $\leftarrow$ POODLE yields a hyponym. Usually, phrases benefit more from hypernyms as they turn specific terms into more general vocabulary that is easier to match with the scene graph labels. Conversely, for scene graph labels hyponym support concepts can be more useful. Therefore, we treat asymmetric relations with different edge directions as two different types of relations when learning. Naturally, this further increases the number of different relation tuples.

5.2.2 Evaluating Candidates

After collecting the candidate concepts and their relation tuples and edge directions, we need to determine which concepts aid the aligner and thus which relations often return useful concepts. The most straight-forward measure would be to count how often a certain relation (or relation tuple) leads to a candidate concept that improves the aligner when placing it in lieu of the original scene graph object or phrase. However, this measure is not ideal for two reasons. First of all, scene graph labels and phrases fulfill different functions. If a phrase is replaced with a candidate, only the alignment of that particular phrase can be improved. If a scene graph label is replaced, however, the change can affect any phrase alignment because the cosine similarity changes for all of them. Thus, the alignment of all phrases has to be evaluated, which is time-consuming. And second, if the candidate concept returns

	hockey player	uniform	ice	hockey stick	puck
skier	**0.28**	0.01	0.31	0.03	0.17
jacket	0.09	**0.44**	0.14	0.11	0.08
snow	0.02	0.06	**0.53**	0.11	0.26
hockey stick	0.14	0.13	0.22	**1.00**	**0.35**
wheel	0.08	0.04	0.13	0.14	0.19

Table 5.2: Cosine similarity adjacency matrix with candidate concept.

the same predictions as the original scene graph label or phrase (both fail or both succeed), it cannot be determined whether the candidate concept could be beneficial. As there is only binary evaluation, smaller improvements cannot be captured. Thus, we require a method that can be applied to both scene graph labels and phrases identically, is easy to compute, returns a numeric value and captures minor changes. The dot product of the similarity adjacency matrix fulfills all these criteria and lets us determine the overall improvement.

The similarity adjacency matrix represents the cosine similarity scores for the embeddings of each phrase and scene graph label pair. An example is shown in Table 5.1, representing the image in Figure 5.2 and the caption *A hockey player in a bright purple uniform skates across the ice with a hockey stick and a puck*. The rows represent the scene graph labels, the columns represent the phrases. We mark the highest similarity score for each phrase, which indicates which scene graph object it is assigned to according to the embeddings similarity. For *hockey player*, *uniform* and *ice* the correct labels are selected. The box *jacket-1* does not completely cover the area of the uniform (pants missing) but the box is a match by evaluation with *IoU*. The phrases *hockey stick* and *puck* are incorrectly aligned to *wheel* and *snow*, respectively.

For each candidate concept, we recompute the adjacency matrix with the candidate concept in place of the original. If the candidate replaces a scene graph label, only the scores in the row for that label change. If the candidate replaces a phrase, only that phrase column is altered. We show an example in Table 5.2, in which the scene graph label *pole* is replaced with *hockey stick*. With this replacement, the cosine similarities are changed and the phrase *hockey stick* is no longer aligned to *wheel*, but to the new candidate *hockey stick*. (Note that the bounding box remains the same even if the label is altered.) Interestingly, the alignment for *puck* is also changed, from *snow* to *hockey stick*. This shows that a change in the scene graph label can affect several alignments.

	hockey player	uniform	ice	hockey stick	puck
skier	1	1	0	0	1
jacket	0	1	0	0	0
snow	0	0	1	0	0
pole	0	0	0	1	0
wheel	0	0	0	0	1

Table 5.3: Ground truth adjacency matrix.

In order to evaluate and compare the two matrices, we need to know which alignments are correct. Therefore, we compute a ground truth adjacency matrix that represents whether the pair of phrase and scene graph label constitute a match when evaluating with IoU. Table 5.3 represents the ground truth adjacency matrix for the example above, 1 representing a correct prediction and 0 an incorrect prediction. Most phrases have one scene graph label for which the prediction is considered correct, except for *uniform*. The boxes for *skier* and *jacket* would both fit with the ground truth box for *uniform* so they are both considered a correct alignment. Note that this is a simplified example as normally there would be a larger number of scene graph labels that are not aligned to any phrase. Furthermore, in this example we assume that there is only one scene graph object per label. Usually, there are several and the largest one is chosen as the prediction. Since the number of scene graph objects per label does not affect the adjacency matrices, we decided to keep this example simple and did not include several objects with the same label.

For both original and candidate concept adjacency matrix, we compute the dot product with the ground truth adjacency matrix by taking their pairwise multiplication. The dot product (weighted sum) increases as the cosine similarities for correct pairs of scene graph label and phrase increase. Since the aligner ranks the scene graph objects by cosine similarity, a higher dot product generally correlates with better alignment performance. The advantage of the dot product is that it captures minor improvements that are not represented in the phrase grounding accuracy as they do not change the final prediction. Despite being minor, they may still affect performance when combined with other improvements and – more importantly – they still contribute information on which relation tuples generally lead to useful candidates. So in order to determine whether a candidate concept is useful, we measure whether the dot product increases. In the example above, the original adjacency matrix yields a score of 1.66. For the adjacency matrix in which *pole* is replaced with *hockey stick* the dot produced increases to 2.54, which is a clear improvement.

Therefore, we consider this extraction of a support concept with the relation tuple ($\leftrightarrow$ *IsRelatedTo,* $\leftarrow$ *IsA*) a success.

We could also assign negative weights (e.g. -0.5) to the incorrect predictions in the ground truth adjacency matrix. This would not only allow measuring whether a new concept increases cosine similarity for correct pairs but also whether it decreases it for incorrect pairs, which is just as valuable. In fact, the candidate *hockey stick* increases the cosine similarity for all phrases, which is not ideal. The dot product difference becomes much smaller if the incorrect prediction pairs have a negative weight of -0.5 assigned. The original version yields -0.238 and the one with the candidate concept yields -0.21, which is still higher but only marginally. We decided not to use negative weights in order to study the effect on the correct predictions alone. Future work could explore the effects of negative weights.

5.2.3 Relation Tuples

Applying this method, we learn with which success rate a relation or relation tuple returns useful support concepts. We collect 983 different relation tuples for extracting support scene graph labels and 816 different relation tuples for extracting support phrases. Unfortunately, the distribution of the relation tuples is sparse. Roughly 24% of all relation tuples occur less than 5 times, only half occur more than 15 times. There are a few outliers, however, that occur up to 166k times, for instance *IsRelated*. This shows that the distribution of ConceptNet relations is highly skewed. We also observe that frequent relations tend to have a lower success rate than infrequent ones. The average success rate is 30%. When discarding relations with fewer than 5 occurrences (which often have a perfect success rate), the average drops to 17%. 73% of all relations have a success rate lower than 50%, for 60% it is even lower than 25%. (These numbers do not significantly vary between the relation tuples for scene graph labels and those for phrases, so we report the average here.) The low success rates could mean that our search radius is too large. Future work should try to extract only candidates that are at most two steps away. As another adjustment one could alter the subgraph generation process, by not including the immediate neighbors of each extracted node at the very end, as the neighbors may introduce concepts that lower success rates.

Tables 5.4 and 5.5 give some examples for relation tuples, their success rate and the number of occurrences. We report the relation tuple with the most occurrences for each range of 10% in the success rate, plus a few interesting cases. The top relation for scene graph label candidates (*IsA* $\leftarrow$, *RelatedTo* $\leftrightarrow$, *IsA* $\rightarrow$) could be interpreted as a connection of the two nodes through a pair of related hyponyms. Note the switch

Relation tuple	Succcess rate (%)	Occurences
IsA ←, RelatedTo ↔, IsA →	86.6	15
HasProperty ←, PartOf →	72.7	11
RelatedTo ↔, DistinctFrom ↔, DerivedFrom ←	66.6	9
HasProperty →, RelatedTo ↔, RelatedTo ↔	52.9	34
RelatedTo ↔, dbpedia/genre →	41.1	107
NotDesires →, IsA →	35.7	28
Antonym ↔, Antonym ↔	26.7	292
Desires →, RelatedTo ↔	17.6	1385
Antonym ↔	14.1	12634
Synonym ↔	10.1	4083
RelatedTo ↔	8.4	188001

Table 5.4: Relation tuples for extracting scene graph label candidates.

in the edge direction of the *IsA* relation. This relation tuple could connect *pet* and *animal* through the hyponyms *dog* and *cat*. The second relation tuple connects the property of a concept with the collection it is part of, for instance (MADE OF) PAPER ← *HasProperty* ← BOOK → *PartOf* → OFFICE/LIBRARY. Interestingly, the relation *RelatedTo* appears in several higher ranking relation tuples but does poorly on its own (8.4%). Since this relation is very vague and links loosely connecting concepts, for instance *dog* with *house*, it is not surprising that it often returns less useful results.

Another observation concerns the relation *Antonym*. On its own, the success rate is 14% but in combination with another *Antonym*, the success rate increases to 26%. Actually, we would expect the opposite of the opposite to be a suitable concept. Unfortunately, the relation *Antonym* does not only connect obvious antonyms (such as *summer* and *winter*) but also links concepts that are alternatives rather than opposites, or sometimes not related at all, for instance DOG ← *Antonym* → CAT ← *Antonym* → CHRISTMAS. The relation *Synonym* performs even worse than antonyms. This can partially be explained by the fact that ConceptNet does not only contain precise synonyms but also more far-fetched relations (e.g. COFFEE ← *Synonym* → CHOCOLATE), just like with antonyms. The more important reason, however, is the lack of word sense disambiguation. This is demonstrated nicely by looking at the synonyms for cat: the synonyms are *feline*, *computerized tomography* and – for unknown reasons – *vomit*. Cat is an ambiguous word that can represent an animal or with capitalized letters a medical imaging procedure (CAT scan). Since the scene graph label *cat* always refers to an animal, it would make sense to remove synonyms for other synsets. However, ConceptNet does not include word

Relation tuple	Success rate (%)	Occurences
NotDesires →, RelatedTo ↔	92.9	14
dbpedia/occupation →, RelatedTo ↔	89.7	39
HasSubevent ←, CausesDesire ←	73.1	26
HasContext ←, HasContext →, HasContext ←	63.4	230
DerivedFrom →, HasContext ←	54.3	103
CapableOf →, CapableOf ←	40.2	206
Desires ←	38.1	805
Antonym ↔	24.3	8570
Synonym ↔	12.9	3801
RelatedTo ↔	11.5	166882
HasContext →	3.3	13838
HasContext ←	31.8	1663

Table 5.5: Relation tuples for extracting phrase candidates.

sense disambiguation, so we cannot differentiate the synonyms, which will introduce unrelated concepts. As shown with the example of *CAT*, ConceptNet does not even mark acronyms.

Table 5.5 with relation tuples for extracting support concepts for phrases features the top relation *NotDesires →, RelatedTo ↔*. The relation *NotDesires* may seem like a relation that does not yield useful results but people are often connected to things they do not want, for example STUDENT → *NotDesires* → DO HOMEWORK. The relation tuple *HasSubevent ←, CausesDesire ←* is also interesting as it often connects objects or scenes with the action they trigger, and the subevent gives more specific aspects of the action, for instance CATCH A WAVE / STANDING ON SURFBOARD ← *HasSubevent* ← SURFING ← *CausesDesire* ← BEACH. If the phrase describes a surfer standing on the surfboard or catching a wave, this relation tuple will lead to the scene *beach*. Another interesting relation tuple is *CapableOf →, CapableOf ←*, which connects two concepts that are capable of doing the same thing. For instance a truck and a ship are both capable of carrying cargo.

The last two rows of Table 5.5 show that the success rate clearly depends on the edge direction: *HasContext →* has a low success rate of 3.3%, while the success rate for *HasContext ←* is 10 times higher. For phrase concepts describing scenes, *HasContext ←* returns elements of the scene, which can be helpful for aligning to scene graph objects, for instance WEDDING ← *HasContext* ← CAKE / BRIDE. For phrase concepts, the relations *Synonym*, *Antonym* and *RelatedTo* perform slightly better than for scene graph labels, probably because the starting concepts are more specific and return fewer and more precise results. Scene graph labels constitute common

words that often connect to many concepts in ConceptNet, among which there are unsuitable ones. They benefit more from relations that return concepts related through hypernyms or hyponyms or the same properties (*IsA, PartOf, HasProperty*), while relations for extracting support phrases often connect concepts through actions (*HasSubevent, CapableOf, NotDesires, Occupation*).

5.2.4 Results and Discussion

The generally low success rates and the low number of occurrences for successful relation tuples present a problem. On average, the relations do not return support concepts that help the aligner, and for many successful relations we do not have enough samples to confirm that they generally improve performance. Nevertheless, we perform experiments incorporating the support concepts and study their effects. For each scene graph label concept and phrase concept in the test set, we retrieve the candidate concepts within a 3-step range in the subgraph. If the relation tuple leading to a candidate concept has a success rate higher than 50%, the concept is accepted as a support concept. If there are several support concepts for the same scene graph label or phrase, we select the one for which the relation tuple shows the highest success rate. We get the word embeddings for the support concept and the original scene graph label or phrase and compute the mean. The support concept is now incorporated in the alignment process, shifting the embedding vector of the original label or phrase.

Table 5.6 shows the results for the application of support concepts on the scene graph labels or phrases separately as well as in combination. For the scene graph labels, we also explore an alternative implementation. Rather than merging the word embedding vectors, we add a new scene graph object that is identical to the original scene graph object, except for the fact that it receives the support concept as a label. This measures whether a support concept performs better when it is not combined with the original label. We conduct the experiments on the best scene graph and aligner version, as determined by the previous chapter, namely the scene graph representation $\mathrm{SG}_{attr+plurals}$ with all aligner extensions activated.

As Table 5.6 shows, the support concepts do not significantly change performance, let alone improve it. The support concepts for phrases perform best among all the support versions but do not surpass the original version. This lack of impact is caused by the fact that only few support concepts are included since most relation tuples have a low success rate and do not meet the threshold of 50%. For 77% of all captions, no support candidates are found for any phrase and for 85% of all images no support concepts are found for any of the scene graph labels. On average, we

	Accuracy (%)
Default	53.85
Support concepts for...	
SG label	53.65
SG label$_{new\ obj}$	53.57
Phrase	53.80
SG label + phrase	53.63
SG label$_{new\ obj}$ + phrase	53.54

Table 5.6: Phrase grounding performance with added support concepts.

find a support concept for 14% of all phrases and 1% of all scene graph labels. The percentage for the scene graph labels is lower because there are on average 64 scene graph objects for each image but only 3.3 phrases per caption. Furthermore, the more specific vocabulary in phrases leads more frequently to support concepts than the common words making up the scene graph labels. For some candidates, the relation tuple does not even occur in the training set so they are discarded because of the missing success rate. This happens on average for 15% of the scene graph label candidates and 20% of the phrase candidates.

As a comparison, we also conducted experiments with a threshold of 75% and 25%, respectively. With a threshold of 75%, the accuracy for any version lies between 53.80% and 53.85%, which is very close to the original version since hardly any scene graph labels and phrases are altered at all. Only 7% of all captions find a support concept for any of their phrases and merely 2% of all images find a support concept for a scene graph label. When lowering the threshold to 25%, naturally more support concepts are included but since they have low success rates, the accuracy is reduced to 52.04%. For 76% of all captions and 56% of all images, a support concept is found for at least one phrase or scene graph label, respectively. Considering the low threshold, these numbers are still not satisfying.

Unfortunately, these results are not conclusive regarding whether or not support concepts could potentially help the aligner, as we simply do not have a sufficient number of high quality support concepts to test with. Since subgraph generation is computationally expensive, our training set consists only of 500 images and thus 2500 captions. Given the large number of different relation tuples and their skewed distribution, the training set is not sufficient to calculate confident success rates for them. The use of a much larger training set (e.g. 5k or 10k images) may help to some extent but in general we expect the problem to persist. There are several ways to tackle this issue. First of all, one could solve the problem of relation tuples

that do not appear in the training data by assigning them the success rate of the most similar relation tuple present. Considering the relation tuple as a string and computing the Levenshtein distance would provide a simple similarity measure. As a means of reducing the large number of different relation tuples, one could cluster them into fewer groups on the basis of the start and return concepts. One could compute the word embeddings and use their distance vector for clustering since similar vector distances represent similar relations when projected on an embedded word. For instance, the distance vectors between *cooking – pan*, *working – computer* and *cleaning – broom* should all be similar as they all represent the relation *tool*. It is, however, not clear whether this approach would hold in a real-world application. Another way of clustering relation tuples would be to group those that return the same concept for a given start concept at least k times, for instance $PartOf \rightarrow$ and $HasA \leftarrow$, or ($IsA \rightarrow$, $Synonym \rightarrow$) and ($Synonym \rightarrow$, $IsA \rightarrow$). Grouping relation tuples could resolve the scarcity issue and thus eliminate relation tuples with few occurrences and questionable high success rates.

Due to their large diversity and skewed distribution, relations may simply not constitute a suitable way of extracting support concepts. Another method would be to select concepts on the basis of word embeddings similarity. One could still exploit the subgraph and only consider nearby or neighboring concepts. Selecting by word embeddings similarity may also reduce the impact of edges connecting nodes that are only loosely related, such as *cat* and *Christmas*, as they are filtered out because of their low similarity. Alternatively, one could incorporate graph metrics for selecting the best candidates, for instance degree, centrality measures such vertex and edge betweenness or personalized PageRank. A combination of the two methods would also be possible. One could also consider adding more than just one support concept for each phrase and scene graph label, as a large number of support concepts may correct for outliers and shift the embeddings vector into the right direction. We refrained from testing this approach as we usually do not have more than one support concept available, if any.

Support concepts may actually improve phrase grounding – if a sufficient number of high quality concepts can be found. Our results do not allow a final conclusion and we leave it to future work to explore better methods for extracting support concepts. When discussing these results and considering new methods, it is important to keep in mind that ConceptNet is not a perfect knowledge base and presents a few challenges in regards to this task. First of all, it is not complete. While *cat* is linked to 7 concepts through the relation *HasProperty*, *mouse* has no edge with that relation at all, although three of the property concepts would apply to mouse as well. Thus, it is difficult to find general relations that apply to many concepts.

Second, ConeptNet contains numerous errors or imprecisions. ConceptNet connects, for instance, many concepts with the relation *Synonym* or *Antonym* that are not true synonyms or antonyms. And third, ConceptNet does not incorporate word sense disambiguation. This introduces related concepts from other word senses and results in unsuitable support concepts. Future work should attempt to integrate existing methods for adding word sense disambiguation to ConceptNet [Chen and Liu, 2011].

5.3 Subgraph Similarity Measures

The support concepts are intended to enhance the existing aligner that is based on the cosine similarity of word embeddings. In this subchapter, we examine other similarity measures that do not depend on the aligner and on word embeddings of scene graph label and phrase. We employ graph metrics that compute the connectedness between a scene graph concept and phrase concept in the subgraph. As a first metric, we use the personalized PageRank, which we already used to generate the subgraph. The personalized PageRank measures the importance of the scene graph concept nodes in regards to a specific phrase concept node, which helps us align the phrase to the correct scene graph label. As a second and third graph metric we include the number and length of the shortest paths between a phrase concept and candidate scene graph concept.

For each phrase concept, we first compute the personalized PageRank for all nodes in the subgraph, using the phrase concept as the starting node. If there are several phrase concepts for the same phrase (for instance *child* as well as *young child*), we compute the PageRanks separately and use the overall mean. For each caption, we save the adjacency matrix representing the personalized PageRanks for all scene graph label and phrase pairs, as demonstrated for the cosine similarity in Table 5.1 and Table 5.2. From this matrix, we can extract the highest ranking scene graph label for each phrase in the caption.

For the other two graph metrics, we compute the shortest path between each phrase and scene graph label pair and save the length of the shortest path as well as the number of shortest paths, as there can be one or several paths of the same (shortest) length, or no path at all. Note that the ideal phrase and scene graph label pair is expected to have a *short* shortest path. A *high* number of shortest paths may be an indicator for a good alignment, though it is neither sufficient nor necessary to make a strong prediction. For both metrics, we create an adjacency matrix as it is done for the personalized PageRank.

5.3.1 Excursion: Pairing Optimization

While taking the highest ranking object for each phrase is a simple and effective way to select alignment pairs, it ignores the possibility that two phrases can be aligned to the same scene graph object. This is true for any similarity measure, including the word embeddings similarity. Table 5.2 demonstrates this on an example: for both the *hockey stick* and *puck*, the highest-ranking scene graph label is *hockey stick*, so they are both aligned to the same scene graph object. So far, we have allowed the aligner to predict the same object for several phrases since every phrase is evaluated individually. In this subchapter, we explore the advantages of first collecting all similarity scores and then finding alignments of non-overlapping pairs.

In the aligner we implemented, we process each phrase individually during alignment, so we do not check whether the selected scene graph object is already aligned to another phrase. However, by computing the similarity scores for all phrases and scene graph labels for each caption and only then selecting the pairs, alignments to the same object can be avoided. In fact, this constitutes an optimization problem known as linear sum assignment or weight matching in bipartite graphs. It concerns itself with problems as the following: *Phrase 1* has two high-ranking candidates named A and B, of which A is ranked slightly higher. *Phrase 2* only has one high-ranking candidate: A. If *Phrase 1* is aligned to A, *Phrase 2* loses its only reasonable candidate and gets aligned to a lower-ranking candidate. Thus, it is better to align *Phrase 1* to B in order to keep A free to be aligned to *Phrase 2*. This way, *Phrase 1* still is aligned to a high-ranking candidate (although not the best) and *Phrase 2* is aligned to its top candidate. The overall loss is reduced.

We employ the linear sum assignment solver[1] from the scipy library for solving this optimization problem. The function returns the optimal alignment pairs based on the similarity adjacency matrix. Since it computes the *minimum* weight matching in bipartite graphs, we have to normalize and invert the similarities (0.0 as highest and 1.0 as lowest similarity score) for the similarity measures based on word embeddings, personalized PageRank and number of shortest paths. The shortest path length similarity does not need to be inverted as lower scores already indicate better pairs.

The goal of our phrase grounding system is to align every phrase to its own individual scene graph object, so pairing using linear sum assignment should be beneficial. It has to be considered, though, that while aligning several phrases to the same scene graph object is theoretically not ideal, one can still achieve a better score in certain cases because of the evaluation by IoU. The baseline used for the phrase grounding

[1] `https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.optimize.linear_sum_assignment.html` (Last accessed: 21.09.2020)

task is a good example, as it shows that aligning to the same large bounding box (i.e. the entire image) still results in 23% accuracy. Thus, one should not necessarily expect the performance to improve when applying this pairing algorithm as it may disable as many accidentally correct predictions as it enables specifically correct ones. If several phrases are aligned to the same large bounding box, enforcing individual predictions may even decrease performance. However, preventing the aligner from grounding several phrases with the same scene graph object is conceptually a reasonable approach and get us closer to solving the actual task of phrase grounding, rather than just optimizing the performance score based on a biased evaluation metric.

Some applications building upon phrase grounding may not even allow overlapping pairs of alignments or at the very least would be confounded by them. For visual question answering, for instance, the system will not be able to deliver the correct answers for questions such as *which animal is bigger* or *which cup is red* if both animals or cups are aligned to the same object.

On the implementation level, linear sum assignment also saves processing time as the similarity adjacency matrix has to be processed only once rather than every column individually. Therefore, we conclude that incorporating this pairing optimization for alignments may reduce the number of accidentally correct predictions, facilitates processing, reduces processing time and prevents confusion in applications using phrase grounding.

5.3.2 Results and Discussion

We compute the phrase grounding accuracy for the new similarity measures using the personalized PageRank and the shortest path by applying the linear sum assignment function on the respective similarity adjacency matrix. The results can be seen in Table 5.7. We also report the accuracy for the original aligner based on embeddings similarities, both with individual alignment and linear sum assignment. We compute the performance for the vanilla aligner as well as the best enhanced version ($SG_{attr+plurals}$ with all aligner extensions activated, although the extensions are not relevant for the other similarity rankings), since the different similarity measures may have different effects on different representations. When two similarity measures are combined, we normalize their scores between 0.0 and 1.0 (0.0 always representing the highest similarity) and take the mean of the two rankings.

As a first step, we analyze the effect of linear sum assignment on the word embeddings similarity measure in comparison to the original version. For both vanilla and

Version	Accuracy (%)	
	$SG_{original+attr}$	$SG_{attr+plurals}$
Embeddings	47.75	53.85
$Embeddings_{linsum}$	47.67	53.64
PageRank	36.40	49.30
$Shortest\ paths_{number}$	2.88	7.94
$Shortest\ paths_{length}$	34.21	42.00
$Shortest\ paths_{number+length}$	29.81	32.46
Embeddings+PageRank	47.05	53.80
Embeddings+$Shortest\ paths_{length}$	47.23	53.45
PageRank+$Shortest\ paths_{length}$	40.88	50.06
All combined	47.76	53.96
All combined (weighted 4:1)	48.02	54.05

Table 5.7: Phrase grounding performance using various similarity measures and their combinations.

enhanced aligner, accuracy drops slightly yet not significantly, as expected. Note that we include the information provided by the aligner extensions about unsuitable candidates by assigning them a score of 1.0, indicating lowest similarity. While the linear sum assignment does not make an impact on performance, it allows us to confidently compare the embeddings similarity measure with the others as the alignments are now all computed the same way.

In general, the results computed on the enhanced scene graph representation are higher, most likely because the plural objects present additional candidates. This demonstrates nicely that the enhanced scene graph representation not only improves alignment for the word embeddings similarity measure but for any of our similarity measures, which suggests that it is a general enhancement.

The personalized PageRank performs surprisingly well. Furthermore, it seems to get a boost from the plural objects in $SG_{attr+plurals}$. The high performance also increases confidence in the subgraphs, which are computed using the personalized PageRank. It is clearly a suitable measure for selecting relevant concept pairs from ConceptNet.

As expected, the number of shortest path does not yield good results. Most pairs have 0 or 1 shortest paths, which does not provide enough information for the aligner to accurately rank the pairs. The shortest path length, on the other hand, yields performances almost at the level of the personalized PageRank. Again, this is a sign of quality for the subgraph. It seems to capture the required information. The two

shortest path similarity measures combined do not outperform the shortest path length similarity on its own, possibly because a larger number of shortest paths occurs more often with long shortest paths. Thus, the number of shortest paths is more suitable for sorting low-ranking candidates rather than high-ranking ones. For the remaining combinations of similarity measures, we only use the shortest path length similarity.

Combining the personalized PageRank similarity measure with the word embeddings similarity measure slightly reduces accuracy, although not significantly. The same is true for the shortest path length similarity measure; it does not improve the version with only word embeddings. This means that the embeddings already represent a strong similarity measure and that the information provided by the personalized PageRank and shortest path similarities is not incorporated well when simply taking the mean of the two measures.

The personalized PageRank and shortest path, although derived from the same subgraph, benefit from each other when they are combined, especially on the original scene graph representation. This is a little surprising as we expect the personalized PageRank to (inversely) correlate with the shortest path length. However, the explicit information provided by the shortest path length similarity may be able to differentiate candidates that have similar personalized PageRanks.

Finally, we combine all similarity measures. Since the embeddings similarity measure proved to be strongest, we double its weight. In other words, we combine the embeddings similarity measure with the already combined measures for personalized PageRank and shortest path length similarity. This yields results similar to the embeddings similarity measure alone, outperforming it by 0.1% on the enhanced scene graph representation. To improve this further, we use the training set to compute the best weights for the two individual measures (embeddings and combination of PageRank and shortest path) and conclude that weighting the rankings 4:1 (in favor of the embeddings similarity) performs best. We evaluate the weighted combined simililarity measures on the test set and it outperforms the original embeddings similarity measure, although not significantly (partially due to the low sample size).

The personalized PageRank and shortest path length similarity measures provide interesting and successful alternatives to the word embeddings similarity measure. Since they are derived from a different source than the word embeddings, they are expected to represent different information, which can complement the word embeddings. Rather than weighting the similarity measures differently, one could try to identify cases for which each similarity measure performs best. For instance, if the word embeddings similarity measure suggests a candidate with much greater

similarity to the phrase than the other high-ranking candidates, the personalized PageRank and shortest path similarity may not be needed. If on the other hand, all word embeddings similarities are low or the highest ranking ones show similar scores, the other two similarity measures could assist. One could also examine which similarity measure works best on which phrase category. Other phrase characteristics such as number (singular/plural phrase), length of the phrase, whether it contains a compound or an adjective may also help in differentiating cases. Future work should analyze the strengths and weaknesses of the different similarity measures and find a better way to combine them in order to get maximum benefit out of each.

5.4 Discussion and Conclusion

We have generated a ConceptNet subgraph for each caption/image pair and extracted support concepts and graph metrics from it to improve phrase grounding performance. The results for each approach can be found in Tables 5.6 and 5.7. The suggested ConceptNet extensions do not result in a significant increase in phrase grounding accuracy. Nevertheless, we have gained some insights into how ConceptNet can and cannot be exploited and what challenges come with it.

The subgraphs have proven to be a suitable way of representing the part of ConceptNet that is of interest for a particular caption. Since the full ConceptNet graph is so large, the subgraphs are necessary to compute metrics and search nodes within a reasonable time. While the subgraph could still be refined and tuned (for instance by not adding neighbors or by imposing a higher minimal Zipf frequency), it provides a good starting point for many extensions based on ConceptNet. The phrase grounding accuracies for the personalized PageRank and shortest path length similarity measures prove that the subgraph does encode information relevant to the task.

The idea of extracting support concepts based on relation tuples was hampered by the diversity and distribution of the relation tuples. Future work will have to find more robust ways for extracting support concepts. Nevertheless, by analyzing the relation tuples in Tables 5.4 and 5.5, we gathered information on the nature of ConceptNet, its strengths and weaknesses. ConceptNet provides information about a great number of concepts and connects them with detailed relations. Unfortunately, one can find many concept pairs that are closely connected in graph but only loosely related to each other, surprisingly also for clearly defined relations such as *Synonym* and *Antonym*. Future work will have to find ways to overcome this issue, either by developing methods that benefit from high recall and do not require high precision,

or by identifying such edges and ranking them lower. For synonyms and antonyms, this could be done by comparing the word embeddings similarity. One could also use co-occurrences extracted from large corpora to rank the edges. That way, the edge between *cat* and *feline* receives a higher ranking than one connecting *cat* and *vomit*. In fact, ConceptNet already does include weights[2] for their edges but they represent the source of the assertion rather than the relatedness of the two concepts, so they are not suitable to resolve this issue.

The graph metrics personalized PageRank and shortest path length have proven to be valuable similarity measures for phrase grounding. While not quite reaching the same performance as word embeddings, they clearly outperform the baseline and when combined reach accuracies close to those for word embeddings similarity. The combination of all similarity measures even achieves a small increase in performance, albeit not a statistically significant one. In order to successfully combine the different similarity measures, one would have to analyze their potential and possibly weight them according to characteristics of the phrase or based on a confidence score. One could also explore other similarity measures derived from subgraph metrics.

Finally, we show that including an optimization algorithm for assigning alignment pairs maintains stable results and although it does not increase performance, it enhances the usefulness of alignments for further applications.

We conclude that ConceptNet is a valuable resource that can be exploited to retrieve information beneficial for phrase grounding. However, ConceptNet and its subgraphs will have to be refined in the future to exclude unsuitable and noise-introducing edges and concepts. The methods for extracting support concepts and combining different similarity measures also have to be explored further. While our experiments reveal some of the challenges of ConceptNet, we also discuss ways for overcoming them. In general, our findings suggest that incorporating ConceptNet can be beneficial and our work provides a foundation for future research on the topic.

[2] `https://github.com/commonsense/conceptnet5/issues/152` (Last accessed: 21.09.2020)

6 Discussion

We sought to improve the scene graph representation of images as well as its application to the phrase grounding task. We pursued three specific aims: analyzing the scene graph representation and phrase grounding system to discover their current limitations, improving the scene graph and the phrase grounding system through a suite of extensions, and incorporating world knowledge based on ConceptNet. Here, we revisit key results and interpret them in the bigger picture.

We tackled the first goal of our thesis in Chapter 3. By analyzing the scene graph representations for Flickr30k images, we learned that the scene graph with its 150 object labels is generally a suitable way of representing images, although it would clearly benefit from a larger number of more fine-grained labels, especially for the categories *other, body parts, instrument* and *scene*. The upper bound for phrase grounding based on the scene graph is 83.3%, which is comparable to object detectors with much larger labels sets, such as *visgen* with 1600 labels and an upper bound of 87.9% [Yang, 2017]. Object detectors generally use labels that are not always suitable for language-related tasks, as they can be ambiguous (*glass* for vessel or visual aid), semantically redundant (*lady* and *woman* or *cap* and *hat*) and derived from different hierarchical levels (*person* > *child* > *boy*). Since object detectors often contribute to multimodal tasks, their label sets should be reviewed and adjusted to facilitate incorporating them into applications that combine the modalities vision and language.

The advantage of the scene graph structure is that it can repesent not only objects but also information about those objects and their relations. The original scene graph only contains positional and possessive relations between objects (e.g. *desk-1 near window-3* or *dog-1 has leg-2*) but it could be extended to contain information about attributes, actions and the scene in general, which would make it an even stronger representation useful for many multimodal tasks.

Two core issues we identified are the fact that the scene graph only represents single objects and that there are often multiple representations for the same people or objects. We tackled both issues as part of the scene graph representation enhancements in Chapter 4. The analysis of the default phrase grounding system showed

that most alignment mistakes are caused either by the missing distinction between singular and plural phrases, the limited label set (e.g. *pole* for *hockey stick*), very specific vocabulary (e.g. *terrier mix*) or abstract and interpreted descriptions (often professions and roles), for instance *band* rather than *people* or *opponent* rather than *man*. To improve alignment quality for such phrases, we developed the ConceptNet extensions in Chapter 5.

We also discovered that the conventional evaluation metric ($IoU >= 0.5$) is biased and does not always provide fair assessments. The evaluation metric favors larger bounding boxes and often classifies predictions as correct, even when it is obvious from the label of the scene graph object that the correct object was actually not identified, for instance *hand-pl* for *guitar* (see Figure 3.13). The baseline provides further proof that the evaluation metric is too forgiving, as every fourth phrase is considered grounded correctly when predicting the entire image as one large bounding box. Since the image frame limits the image size, larger objects naturally overlap more frequently than smaller ones, which increases their IoU. Thus, by slightly enlarging the predicted boxes one could probably achieve an increase in accuracy, without any improvement at all in the actual quality of the predictions. Thus, performance boosts caused by promoting larger bounding boxes must be taken with a grain of salt (for example the introduction of plural objects). On the other hand, a decrease in performance caused by the removal of candidates with large bounding boxes, as it occurs for object reduction in order to reduce multiple representations, may not actually indicate a lower quality of the alignments. Naturally, if the evaluation metric is biased and does not accurately reflect the quality of phrase grounding, it becomes difficult to improve the system and verify the increase in performance for a given extension. It is important to note that the transparency of unsupervised systems enables an exact examination of how a prediction is computed, which allows such issues to be identified. Supervised models lack this option, so it may escape notice that they are affected as much or more by a biased evaluation metric.

This issue is not unique to phrase grounding. Blagec et al. [2020] suggest that many performance metrics used for comparing AI models on benchmark datasets do not reflect performance adequately as they do not cover all relevant performance characteristics. Future work will have to find a better evaluation metric (or set of metrics) for phrase grounding that is not biased towards larger bounding boxes and better reflects whether the object(s) mentioned in the phrase are actually correctly identified. It could be argued that for many applications it does not matter whether the predicted box is the result of correct alignment or a more or less accidental match, as long as it roughly covers the region in question. However, one should consider that there are applications that require high-precision boxes as more information needs

to be extracted from them, for instance for visual question answering, or tasks for which finding a matching box is not the ultimate goal, for instance sentence-based image retrieval. Even if the applications are not affected by slightly smaller or larger bounding boxes, an adequate evaluation metric is essential for efficiently optimizing and accurately evaluating the performance of phrase grounding, especially when comparing different models, as some may benefit from the biased evaluation metric and others may experience detriment.

It is not trivial to design an alternative general evaluation metric that is applicable to all systems. The word embeddings similarity measure could evaluate how well the predicted label fits the phrase but this may not always return desirable results. In fact, one of the challenges of phrase grounding is that phrase and object label are not necessarily similar so using this measure for evaluation is not recommended. Since different models work with different object detectors and thus different label sets, this evaluation metric would also not be fair. Regarding size bias, when working with segmentation masks rather than bounding boxes the pixel IoU may provide a more adequate evaluation metric, as it is more precise and eliminates the issue of "empty" corners. However, many phrase grounding models are based on object detectors that return bounding boxes rather than segmentation masks and the Flickr30k Entities dataset only provides ground truth boxes. A relatively easy way to adapt the evaluation metric would be to introduce a penalty for large boxes, especially when they reach the image frame. The IoU should be higher for such cases in order to be considered a correct prediction. One could also include information about how many bounding boxes of scene graph objects would match the ground truth. If there are five different objects that match the phrase in question, the IoU has to be higher than for cases in which there is only one matching scene graph object. Whether such adjustments would lead to a more accurate performance metric or whether it would unintentionally reduce the number of correct and reasonable predictions would have to be explored in future work.

We addressed the second goal of our thesis in Chapter 4. We enhanced the scene graph representation with plural objects and attributes about color and whether a person is found in the background of the image. We also attempted to remove multiple objects representing the same instance. The inclusion of plural objects is an obvious success. Even without the corresponding extension of distinguishing singular and plural phrases during alignment, the new representation achieves an increase in accuracy of nearly 2%. It also boosts the performance for other similarity measures, for instance the personalized PageRank and the shortest path length similarity measures (see Chapter 5), which suggests that it is a general enhancement of the scene graph. As mentioned above, the plural objects introduce larger objects

that are favored by the evaluation metric, so a fraction of the performance increase is possibly caused by the biased evaluation metric. However, the aligner extension for number distinction only allows plural phrases (16%) to be aligned to plural objects, which prevents unfair exploitation of large plural objects. We also generated plural objects in a controlled way. Therefore, we believe that the improvement achieved by adding plural objects is largely genuine.

We also attempted to remove multiple representations of the same object from the scene graph. Such superfluous objects generally decrease the quality of the scene graph, both because of the multiple instances and the frequently incorrect labels for people objects. They can also interfere with the aligner extensions and plural object generation, especially for people. Interestingly, phrase grounding performance is less affected by the multiple representation issue than one might expect. Since the bounding boxes of these objects are only partially overlapping, merging them into plural objects generates large candidate boxes, which the evaluation metric favors. The lack of impact of these multiple representations is shown by the phrase grounding performance on the reduced scene graph representation. Removing 3% of the scene graph objects, which corresponds to a 20% reduction of people objects, halved the number of superfluous people objects on a manual evaluation set. However, the phrase grounding performance is decreased, rather than increased because higher recall is more effective than higher precision. Thus, a higher quality scene graph representation does not necessarily improve the performance of phrase grounding, which is another consequence of the biased evaluation metric.

The approach we followed for merging bounding boxes can be refined and extended in order to reduce a larger number of superfluous boxes and promote correctly labeled objects. One should also test other object detectors and evaluate whether they cause multiple representations on the same scale. One could also reduce the number of relationship triples used to generate the scene graph, although the analysis has shown that even among the 20 most confident triples there are multiple representations, and using even fewer triples would result in uninformative scene graphs. A clean scene graph without multiple representations of the same object should ultimately improve performance, especially combined with our aligner extensions. It will also improve the quality of the scene graph representation in general, making it more useful for other applications.

As part of the second research goal, we also introduced three different extensions for the aligner. The number distinction as a complement for the plural objects increases the performance by 4% and is clearly the strongest of the extensions. The color attribute does not achieve a significant performance boost as it only affects few phrases. Nevertheless, it constitutes a proof-of-concept showing that attributes can

help the aligner, especially for differentiating objects with the same label. The color and background attribute also enhance the scene graph with additional information. The extensions for grounding people phrases, namely the foreground preference and label preference, achieve a small, yet significant improvement when combined. The label preference will have to be extended with automatically extracted candidate label sets for people and non-people objects in order to make a bigger impact on performance. The combination of different extensions outperforms the individual extensions, which proves that they do not inhibit each other's effect but benefit from each other, as the different extensions filter out different unwanted candidates.

The third search goal concerned itself with the integration of external world knowledge derived from ConceptNet into the phrase grounding system. ConceptNet turned out to be a challenging resource for several reasons. It is less structured than WordNet and a concept can have any number of edges of each type, which often returns too many or no results at all. The relations are not always meaningful, as loosely related concepts can be connected. There is also no word sense disambiguation. Finally, ConceptNet is so large that processing is only possible when working on subgraphs.

We generated subgraphs to enable and facilitate processing the information in the ConceptNet graph. Despite potential for improvement, we consider the subgraph generation a success as the graph metrics derived from it were useful for phrase grounding. Whether the subgraph yields good support concepts remains unclear, as our algorithm for extracting support concepts through relation tuples proved unsuitable due to their skewed distribution. We were not able to extract a sufficient number of high-quality support concepts to measure their effect on phrase grounding, although we suggested several options for improving support concept retrieval in future work.

The personalized PageRank and the shortest path length similarity extracted from the subgraph achieved promising results as alternative similarity measures for phrase grounding with 49% and 42%, respectively. Their combination shows that they benefit from each other (50%). However, combining the new similarity measures with the word embeddings similarity measure does not significantly outperform the classical aligner using word embeddings alone. Nevertheless, future work could try to get the most out of each similarity measure by using confidence scores and case-specific applications of the different rankings, which may well improve phrase grounding above the current level of our system.

Over the course of this thesis, we managed to improve the phrase grounding system by a significant margin. The vanilla version on the original scene graph represen-

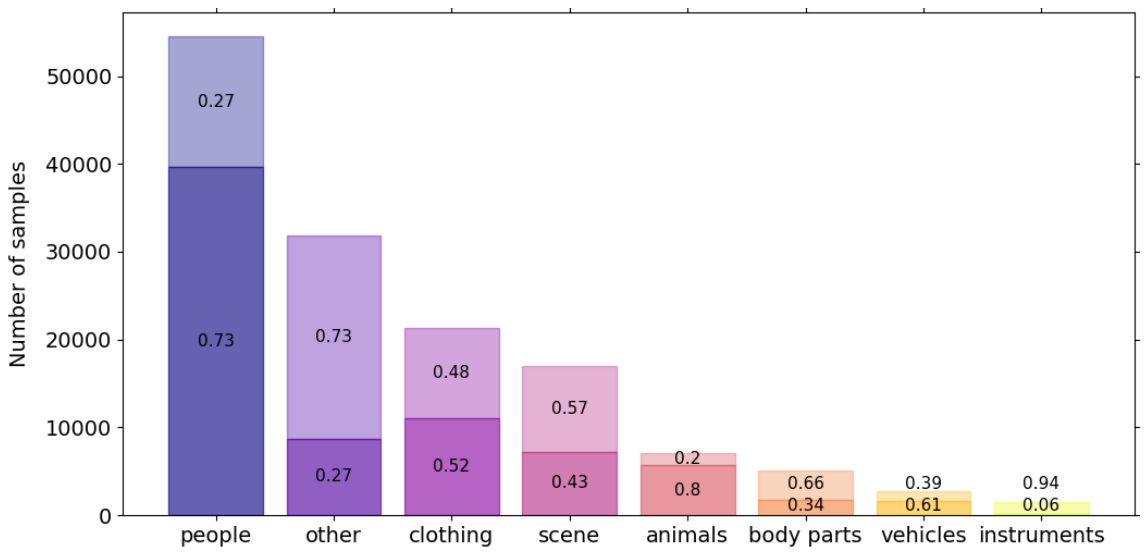


Figure 6.1: Phrase grounding accuracy by category for improved system.

tation yields 47.9% accuracy. Our best system employing the scene graph representation with plural objects and all extensions activated achieves 53.7%. Adding the weighted personalized PageRank and shortest path length similarity measures, we ultimately reach 54.0%, which constitutes a 6% increase in performance over the original system. Figure 6.1 shows the improved performance by category. The originally best-performing categories (see Figure 3.11) *animals*, *people*, *vehicles* and *clothing* are the ones that benefit most from the enhancements, probably because of their high Flickr30k coverage, which indicates that there exists a correct bounding box for them in the scene graph. The categories *animals* and *people* improve by 9%, *clothing* by 7% and *vehicles* by 5%. Plural phrases often describe people and animals so we assume that those categories benefit most from the plural objects. The category *people* is further pushed by the extensions for grounding people objects. The categories *other* and *body parts* increase by 4%, *instruments* by 2% and *scenes* by 1%. Since these categories have a lower Flickr30k coverage and in the case of the latter two no scene graph labels describing them, it is not surprising that they improve less. These are categories that would likely benefit most from another object detector with more fine-grained labels. Ultimately, all categories are improved by the enhancements, which suggests that they are fairly general.

Note that while our system is generally regarded as an unsupervised model, we included learning methods in some extensions. For object reduction, for instance, we learn the thresholds for region overlap and neighbor overlap by parameter screening on a training set. To extract support concepts, we learn the success rates of the

relation tuples. The actual phrase grounding system, however, does not include any training and thus we would still describe it as an unsupervised system.

Finally, we compare our system with the complete model by Parcalabescu and Frank [2020]. In order to better study the individual enhancements, we only worked with their base system, omitting the steps for contextualizing the image and language representation, such as the incorporation of scene graph relationships, knowledge injection through WordNet and Open Images and the WordNet path similarity measure. Thus, our system could technically be described as a *bag of objects* model, as the vanilla system does not utilize information extracted from the scene graph relations. However, we employ the neighbor overlap as a threshold measure for object reduction and make use of the additional attribute nodes for color and background, which compare different scene graph objects with each other. Future work should attempt to incorporate more information provided by the scene graph structure, not just by the objects. An example for such an extension involves comparing the syntactic structure of the caption with the scene graph structure. If the nominal phrase *a man* in a caption has the prepositional phrase *with a hat* dependent on it and the scene graph contains the relation *man-1 wearing hat-1*, we can exploit the structural information to achieve better alignments. If there are several men in the image, we can align the phrase *a man* to the scene graph object that is connected to *hat-1*. The same applies to the case in which the phrase describing the man is more abstract (e.g. *a musician*); the connection to *hat* allows correct alignments even if the phrase and scene graph object do not have a high similarity ranking.

Since Parcalabescu and Frank [2020] evaluated their system using different object detectors and on a different test set, direct comparisons to our system are not possible. However, we can compare the trends. Their best system improves the *bag of objects* approach [Wang and Specia, 2019] from 46.08% to 47.92% using the *tfoid* object detector and from 56.30% to 57.08% using *visgen*, reaching an accuracy increase of nearly 2%. The WordNet path similarity has the highest impact. Our system employing the scene graph generator by Zellers et al. [2018] works with only 150 labels but has a similarly high upper bound as *visgen* (83.3%). Our enhancements improve the vanilla *bag of objects* system from 47.9% to 54.0%. While a direct comparison is not possible due to different object detectors, it is likely safe to conclude that our enhancements have a bigger impact on the system, since our increase in accuracy is 6% while for Parcalabescu and Frank [2020] it is less than 2% (on two different object detectors). The combination of their full system with our enhancements will probably outperform both individual systems, as it combines contextualization for image and language representation and our enhancements. We assume that the WordNet path similarity would help our system the most as it introduces missing hierarchi-

cal knowledge. On the other hand, the system by Parcalabescu and Frank [2020] will most likely benefit from the plural objects and the number distinction, as they currently do not include methods for distinguishing singular and plural phrases and objects. Since the contextualization methods and our enhancements are all fairly independent from each other, building a combined system should be feasible.

It would also be interesting to measure the performance boost to our system when using a more fine-grained object detector. More fine-grained labels could also have a positive effect on the ConceptNet extensions, which sometimes struggle with very common words. The authors of Parcalabescu and Frank [2020] told us the unreported accuracy (47.52%) for their best system using the same of object detector as we did. Unfortunately, the accuracy for the default system could not be retrieved so the improvement rate cannot be computed. However, this score allows to compare the impact of a different object detector. When incorporating *visgen* as an object detector (1600 labels) that shows a slightly higher upper bound (87.9%) than ours, we can expect a performance improvement up to 10%.

Although not directly evaluated against other phrase grounding systems, we expect that our system, especially when using a more fine-grained object detector, would outperform all weakly supervised models and some strongly supervised models, probably with the exception of QRC-Net [Chen et al., 2017b] and VisualBERT [Li et al., 2019], as our system is very similar in structure to Parcalabescu and Frank [2020] and they achieve such results. This shows that our unsupervised approach could rival complex supervised models that require large amounts of training data.

Parcalabescu and Frank [2020] also perform a stress test on the sentence-based image retrieval task: for a given caption, the correct image has to be selected among 1000 candidate images. They reach a recall (R@1) of 15.3, which cannot compete with recent supervised models (58.2% [Lu et al., 2019]) but shows the potential of their system on other applications. We did not evaluate our system on sentence-based image retrieval but it would be interesting to examine whether some of our extensions that show little or no effect on phrase grounding would improve the performance on this task, for instance for label preference and the removal of multiple representations. In fact, the sentence-based image retrieval task could function as a sanity check for new evaluation metrics for phrase grounding. On this task, finding a box that matches the ground truth is no longer the ultimate goal, at least not for the incorrect candidate images. Finding boxes that do not actually match by label would even reduce performance as they promote incorrect candidates. Thus, a good evaluation metric for phrase grounding should not only reward the system when it finds a suitable box, it should also punish it when finding an unsuitable one, which is exactly captured by the sentence-based image retrieval task.

7 Conclusion

In this thesis, we have improved the phrase grounding system introduced by Parcalabescu and Frank [2020] that uses the scene graph as a representation of the image. We have analyzed the scene graph representation and extrapolated two core issues: multiple representations of the same object and no objects representing multiple instances. We have also suggested methods for enriching the scene graph with further information, for instance attributes and actions. We have examined the performance of the phrase grounding system and identified frequent error sources for incorrect alignnments.

We have enhanced the scene graph representation by adding plural objects and removing superfluous objects in order to tackle the identified core issues. We have also incorporated two attributes: color and foreground/background dinstinction. To improve the phrase grounding system, we have developed three different extensions that filter candidate objects based on color, whether the phrase and object represent multiple instances, and the age, gender and position of people. With these enhancements, we have achieved an increase in phrase grounding accuracy of 6%. For each method, we have also proposed ways to further improve the system in future work.

Furthermore, we have integrated methods for including external knowledge from ConceptNet. We have generated a subgraph for each caption, representing only relevant concepts. We have attempted to extract support concepts for phrases and scene graph labels based on learned success rates of relation tuples but this approach was not successful due to their skewed distribution, so we proposed ways to overcome this issue in future work. We have also extracted two alternative similarity measures from the subgraph, the personalized PageRank and shortest path length between candidate pairs, and achieved promising results on the phrase grounding task. Although the word embeddings similarity is not outperformed when combined with the new similarity measures, we have proposed ways to weight the different similarity measures in order to get the most benefit out of each.

On a more general level, we have discovered that the conventional evaluation metric ($IoU >= 0.5$) does not always provide an adequate evaluation as it is biased towards larger bounding boxes. It classifies predictions as correct even if their label betrays

that they do not represent the object described in the phrase. The evaluation metric favors recall over precision, which can be problematic for applications that require suitable labels and precise bounding boxes. Having a biased evaluation metrics also makes it harder to verify improvements and compare different models with each other, but developing a new one is not trivial either.

Finally, we have compared our results with the ones achieved in Parcalabescu and Frank [2020] and concluded that our enhancements most likely lead to more substantial improvements over the *bag of objects* system than theirs when evaluated using the same object detector. A combination of their contextualized representation of image and language and our enhancements for improving scene graph quality and tackling specific phrase grounding challenges would probably outperform the individual systems. Incorporating a more fine-grained object detector also seems promising. Since Parcalabescu and Frank [2020] already outperform all weakly supervised and some strongly supervised models, our combined system could further challenge current supervised architectures, which are highly complex, lack transparency and require large training data, yet do not substantially outperform this unsupervised approach.

Phrase grounding is a challenging multimodal task that requires information from both image and caption as well as external knowledge in order to be solved. In this work, we have analyzed the challenges and opportunities of an unsupervised system and have shown that carefully designed methods based on a thorough analysis can be more valuable than large amounts of training data. Furthermore, we have demonstrated that the scene graph constitutes a powerful image representation that when further enhanced with attributes and relations provides an invaluable data structure for multimodal tasks that require a link between the visual and textual modality.

References

M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet Project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, 1998.

K. Blagec, G. Dorffner, M. Moradi, and M. Samwald. A Critical Analysis of Metrics Used for Measuring Progress in Artificial Intelligence. *arXiv preprint arXiv:2008.02577*, 2020.

Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. HICO: A Benchmark for Recognizing Human-Object Interactions in Images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1017–1025, 2015.

J. Chen and J. Liu. Combining ConceptNet and WordNet for Word Sense Disambiguation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 686–694, 2011.

K. Chen, R. Kovvuri, J. Gao, and R. Nevatia. MSRC: Multimodal Spatial Regression with Semantic Context for Phrase Grounding. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 23–31, 2017a.

K. Chen, R. Kovvuri, and R. Nevatia. Query-Guided Regression Network with Context Policy for Phrase Grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 824–832, 2017b.

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*, pages 4042–4050, 2018.

X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.

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 International Conference on Computer Vision*, pages 2601–2610, 2019.

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*, pages 248–255. Ieee, 2009.

P. Dogan, L. Sigal, and M. Gross. Neural Sequential Prase Grounding (seqGROUND). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4175–4184, 2019.

C. Fellbaum. WordNet: An Electronic Lexical Database, 1998.

R. Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

S. Guadarrama, E. Rodner, K. Saenko, N. Zhang, R. Farrell, J. Donahue, and T. Darrell. Open-Vocabulary Object Retrieval. In *Robotics: Science and Systems*, volume 2, page 6, 2014.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

R. Hinami and S. Satoh. Discriminative Learning of Open-Vocabulary Object Retrieval and Localization by Negative Phrase Augmentation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2605–2615, 2017.

M. Honnibal and I. Montani. spaCy 2: Natural Language Understanding with Bloom Embeddings. *convolutional neural networks and incremental parsing*, 7 (1), 2017.

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*, pages 4555–4564, 2016.

J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image Retrieval Using Scene Graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

A. Karpathy, A. Joulin, and L. F. Fei-Fei. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

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)*, pages 787–798, 2014.

G. R. Kress. *Multimodality: A Social Semiotic Approach to Contemporary Communication*. Taylor & Francis, 2010.

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*, 123(1):32–73, 2017.

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.

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*, pages 21–37. Springer, 2016.

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*, pages 13–23, 2019.

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*, pages 3111–3119, 2013.

S. Minaee and A. Abdolrashidi. Deep-Emotion: Facial Expression Recognition using Attentional Convolutional Network. *arXiv preprint arXiv:1902.01019*, 2019.

L. Parcalabescu and A. Frank. Exploring Phrase Grounding Without Training: Contextualisation and Extension to Text-Based Image Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 962–963, 2020.

D. Paul and A. Frank. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1:3671 – 3681, 2019.

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*, pages 2641–2649, 2015.

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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1928–1937, 2017.

B. A. Plummer, P. Kordas, M. Hadi Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik. Conditional Image-Text Embedding Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018.

J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

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*, pages 779–788, 2016.

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*, pages 817–834. Springer, 2016.

J. Ruppenhofer, M. Ellsworth, M. Schwarzer-Petruck, C. R. Johnson, and J. Scheffczyk. *FrameNet II: Extended Theory and Practice*. Institut für Deutsche Sprache Bibliothek, 2006.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet Large Scale Visual

Recognition Challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

A. Sadhu, K. Chen, and R. Nevatia. Zero-Shot Grounding of Objects from Natural Language Queries. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4694–4703, 2019.

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *In ICLR*, 2013.

C. Silberer and M. Pinkal. Grounding Semantic Roles in Images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2616–2626, 2018.

R. Speer and C. Havasi. Representing General Relational Knowledge in ConceptNet 5. In *LREC*, pages 3679–3686, 2012.

R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

R. Speer, J. Chin, A. Lin, S. Jewett, and L. Nathan. LuminosoInsight/wordfreq: v2.2. Oct. 2018. doi: 10.5281/zenodo.1443582. URL `https://doi.org/10.5281/zenodo.1443582`. Last accessed: 21.09.2020.

D. Teney, L. Liu, and A. van Den Hengel. Graph-Structured Representations for Visual Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2017.

M. Tkalcic and J. F. Tasic. *Colour Spaces: Perceptual, Historical and Applicational Background*, volume 1. IEEE, 2003.

J. Wang and L. Specia. Phrase Localization Without Paired Training Examples. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4663–4672, 2019.

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*, 41(2):394–407, 2018.

M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng. Structured Matching for Phrase Localization. In *European Conference on Computer Vision*, pages 696–711. Springer, 2016.

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*, pages 5945–5954, 2017.

D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

J. Yang. Faster-RCNN trained on Visual Genome. 2017. URL `https://github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome`. Last accessed: 21.09.2020.

B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human Action Recognition by Learning Bases of Action Attributes and Parts. In *2011 International conference on computer vision*, pages 1331–1338. IEEE, 2011.

R. A. Yeh, M. N. Do, and A. G. Schwing. Unsupervised Textual Grounding: Linking Words to Image Concepts. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference Over Event Descriptions. *Transactions of the Association for Computational Linguistics*, 2: 67–78, 2014.

R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

Y. Zhang, L. Yuan, Y. Guo, Z. He, I.-A. Huang, and H. Lee. Discriminative Bimodal Networks for Visual Localization and Detection with Natural Language Queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 557–566, 2017.

F. Zhao, J. Li, J. Zhao, and J. Feng. Weakly Supervised Phrase Localization with Multi-Scale Anchored Transformer Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5696–5705, 2018.