Master Thesis

ICA-6861008

Using Natural Language Inference to Perform Visual Inference: the Case of Quantified Noun Phrases

Jana Lipping

August 2021

Supervisors:
Kees van Deemter
Guanyi Chen
Ad Feelders

Department of Information and Computing Sciences

Utrecht University

Abstract

Evaluation of quantities in visual data remains one of the biggest challenges in the area of Visual Inference. We explore a novel approach to reasoning about quantities in visual contexts using the tools of Natural Language Inference, working with textual descriptions of visual scenes. Based on a complete description of a simple geometrical scene, we try to predict if a quantified statement about objects in this scene follows from the description. We test an LSTM-based neural network architecture on this task and examine the generalization ability of the model.

Contents

Chapter 1

Introduction

Modelling visual intelligence is a major subject of research in artificial intelligence. Visual reasoning, or the ability to accurately explain the content of visual scenes, is an essential skill of any intelligent system. The complex dual nature of this problem that involves acquiring the ability to understand visual data and also to produce correct descriptions in natural language puts it at the nexus of computer vision and natural language processing. Visual Inference (VI) is a challenging subtask of visual reasoning that consists in learning to detect whether a statement about the content of an image follows from the image. As the goal of VI is modelling deep understanding of visual data, an important step is designing models that can assess relations between sets of objects in visual scenes. This is done through quantitative reasoning. Evaluating quantities in visual contexts has largely been studied via counting objects in images in computer vision or Visual Question Answering (VQA). Such research primarily focuses on numbers. Application of diverse linguistic quantitative phenomena to visual scenes has not been sufficiently studied. Automatically learning the logic that underlies varied quantification mechanisms is the subject of this thesis.

Quantifiers, or generalized quantifiers (following the definition by Barwise and Cooper (Barwise and Cooper, 1981)), include determiners like ‘many’, ’most’, ’all’, ’no’ and ’some’, numbers, proportions and composite expressions like ’at least 3’. In natural language quantifiers can usually be seen alongside nouns. Combined with nouns, they form quantified noun phrases, such as ’most people’. Quantifiers are studied in many different areas such as psychology, linguistics and logic. There is a general consensus that a quantified statement like ”All triangles are blue”, that can be described by the formula Q(A,B), where Q is ’all’, A is ’triangles’ and B is ’blue’, captures information about a relation between sets A and B (Geurts et al., 2010).

Different quantification mechanisms differ in their logical properties as well as how they are processed by humans. When it comes to studying quantification mechanisms, it is important to recognize the difference between proficiency with numbers, or numeracy, the ability to assign the correct non-numerical quantifier such as ’many’ or ’few’ to a scene, and the proportional estimation skill. Research shows that children do not learn and estimate numerical and non-numerical quantifiers the same way. Children learn numbers around the same age as they learn the meaning of quantifiers like ’some’, however, they do not learn the correct interpretation of such quantifiers until later (Hurewitz et al., 2006); the evaluation of proportions is the most advanced skill that is learned much later (Hartnett and Gelman, 1998). These findings prove that humans process different methods of quantification differently and provide the intuition for modelling human quantitative reasoning.

While previous research has largely focused on generating sentences depicting the content of visual scenes based on images, we aim to investigate the potential value of using textual data to learn to quantify objects in visual contexts. We will look at learning quantifiers through the lens of Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE) (Dagan, Glickman, and Magnini, 2005). NLI is a challenging task that is concerned with learning to detect if a premise sentence entails a hypothesis sentence. The general goal of NLI is modelling deep understanding of sentences and logical relationships between them. There have been many advances in NLI in the recent years (S. R. Bowman et al., 2015; A. P. Parikh et al., 2016; Q. Chen et al., 2016). Few studies in NLI have focused on quantification, which serves as motivation for our project. The objective of this project is to evaluate whether neural networks can learn to be able to predict, given a complete description of a visual scene, whether a statement about objects in the scene is true or false. In this way our task can be seen as a potential second component of a Visual Inference framework, where the content of visual scenes is first translated into textual form.

Research question. The goal of the thesis is to build a machine learning model that, given a simple geometrical scene, is able to tell if a textual description of it is true or false (e.g., 'Most circles are blue'.) More precisely, the model is supposed to predict if there is an inference relationship between a full description of a scene and a statement about one type of objects present in the scene. For example, if the initial description of an image containing 4 circles, 3 of which are green and 1 of which is red, and 4 blue triangles, is: 'there are 3 green circles, there is 1 red circle, there are 4 blue triangles', then the model should be able to tell that '4 objects are circles' is true. Complete image descriptions of this kind can be generated with the tools of computer vision, which we describe in more detail in the next chapter.

To summarise, in this thesis we will try to answer the following **research question**:

Based on a complete description of a visual scene containing geometrical objects, are simple neural networks able to learn to infer which quantified statements about the objects in the scene are true and which are false?

Contribution. In this thesis to answer the research question we formulate the task of Quantified Natural Language Inference for Visual Inference, propose a neural network model for the task, present an experimental framework and generate datasets to conduct the experiments.

Chapter 2

Background/Related work

2.1 Quantifiers

2.1.1 Generalized Quantifier theory

The study of quantifiers dates back to the foundations of logic and Aristotle who studied the properties of the four basic quantifiers 'all', 'no', 'some', 'not all'. 'Generalized quantifiers' is a term that reflects the fact that quantifiers are a generalization of the logic quantifiers $\forall$ and $\exists$. Semantically quantifiers are analysed within Generalized Quantifier theory. Generalized Quantifier theory was initially established by Mostowski (Mostowski, 1957) and was further developed by Barwise & Cooper (Barwise and Cooper, 1981).

In order to judge if a quantified expression is true or not, it is necessary to define its truth conditions. The first to ask the question of what makes a sentence formally correct was the logician Alfred Tarski (Tarski, 1933). In model-theoretic semantics by Montague the truth of a statement is assessed with respect to a model (Montague, 1973). A model M is a domain/universe, that is denoted by D_M, coupled with a function $[\![\cdot]\!]_M$ that associates syntactic constituents with their meaning, called a semantic value. The semantic value of complex expressions is defined based on the semantic values of their atomic parts and rules for assigning semantic values to expressions that contain them. It is possible to determine the truth value of sentences with quantifiers in first order predicate logic, however, the syntax of predicate logic is so different from the syntax of natural languages that it is hard to translate sentences in English to predicate logic. Because of these challenges Generalized Quantifier theory emerged as a framework of analysing quantified statements. It was developed with the goal to bridge the gap between the predicate logic approach to quantifiers semantics and the syntax of quantified noun phrases in natural language. In Generalized Quantifier theory sentences with quantifiers express relations between sets. In model-theoretic notation:

$[\![\text{All objects are red}]\!]_M = \text{True iff } D_M \subseteq [\![\text{red}]\!]_M$

'All objects are red' is true in the model M iff the set of objects in the model is a subset of the set of red objects in the model.

$[\![\text{Some objects are red}]\!]_M = \text{True iff } D_M \cap [\![\text{red}]\!]_M \neq \varnothing$

'Some objects are red' is true in the model M iff the intersection of set of objects in the model and the set of red objects in the model is not empty.

$[\![\text{No objects are red}]\!]_M = \text{True iff } D_M \cap [\![\text{red}]\!]_M = \varnothing$

'No objects are red' is true in the model M iff the intersection of set of objects in the model and the set of red objects in the model is empty.

Quantified noun phrases define sets of sets:
$[\![\text{All objects}]\!]_M = \{X : D_M \subseteq X\}$
$[\![\text{Some objects}]\!]_M = \{X : D_M \cap X \neq \varnothing\}$
$[\![\text{No objects}]\!]_M = \{X : D_M \cap X = \varnothing\}$
Predicates define sets. Therefore sentences with quantified noun phrases are interpreted in the following way:
$[\![\text{NP Pred}]\!]_M = \text{True iff } [\![\text{Pred}]\!]_M \in [\![\text{NP}]\!]_M$
'Some objects are red' is true in the model iff the set of red objects in the model belongs to the set of sets for which their intersection with the domain of the model is not empty.

Now we can define the truth condition for entailment: a sentence ψ entails a sentence ϕ if ϕ is true in all models in which ψ is true.

In our data premises serve as descriptions of the universe of a model. We consider only the objects described in the premise to be present in the visual scene. The truth of the hypothesis is established in the model with the universe described in the premise.

2.1.2 Types of quantifiers

Several types of generalized quantifiers can be distinguished. Some of them are:

1. **Aristotelian**. These are standard quantifiers 'no', 'some', 'all', 'not all'.

2. **Cardinal**. Example of a cardinal quantifier is 'more than 2'. The applicability of this type of quantifiers depends on the cardinality of sets.

3. **Proportional**. Examples of proportional quantifiers are 'most', '10%'. Whether this type of quantifiers hold or not depends on the proportion of objects having a certain property.

4. **Vague**. Example of a vague quantifier is 'many'. This kind of quantifiers is different in that a threshold needs to be defined in order to judge if these quantifiers hold.

2.1.3 Semantic complexity of quantifiers

Quantifiers vary in the number of operations required to perform to establish the truth value of sentences containing them. Van Benthem proposed to classify quantifiers in terms of automata that compute their truth value (Van Benthem et al., 1986).

1. **Aristotelian quantifiers** 'no', 'some', 'all', 'not all'. The computation of the truth value of these quantifiers requires checking a condition for all objects in the model. For example, to evaluate the truth of the sentence 'All objects are red' it is necessary to verify if each object has the color red. For that we need a 2-state automaton that starts in the 'true' state and stays in it until an object that is not red is discovered, in which case it goes to the 'false' state and then stays there no matter what color the remaining objects have. This automaton is shown in Figure 2.1.

2. **Cardinal quantifiers**. To find the truth value of cardinal quantifiers like 'more than 3', '5' we also need to check the color of each object in the model. To determine the truth value of the sentence 'More than 3 objects are red', we need to find at least 4 red objects. We start in the 'false' state and move to the next 'false' state upon discovering a red object, this happens 3 times. If the fourth red object is discovered, we move to the 'true' state. This automaton is shown in Figure 2.2.

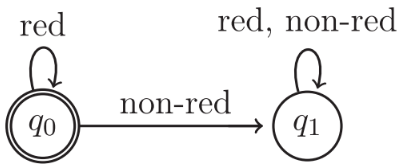


Figure 2.1: The 2-state automaton characterizing Aristotelian quantifiers (Szymanik and Thorne, 2017)

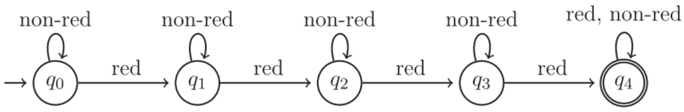


Figure 2.2: The 5-state automaton for computing the truth value of the quantifier 'more than 3' (Szymanik and Thorne, 2017)

3. **Proportional quantifiers**. The computation of the truth value of proportional quantifiers like 'fewer than half' requires comparing the size of the set of objects with the considered property and the set of remaining objects. To accomplish this, a push down automaton (with a stack) is necessary. To compute the truth value of the sentence 'Fewer than half of objects are red' we need to save information about the previously checked objects and cancel out pairs of red and non-red objects. This automaton is shown in Figure 2.3.

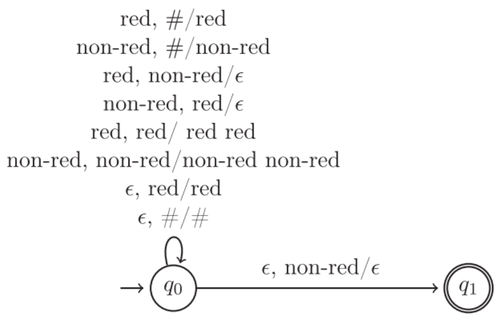


Figure 2.3: The automaton for computing the truth value of the quantifier 'fewer than half' (Szymanik and Thorne, 2017)

2.2 Learning the meaning of quantifiers in visual contexts

The interplay between visual and textual modalities is explored in numerous tasks, such as image captioning (Hodosh, Young, and Hockenmaier, 2013), visual question answering (Antol et al., 2015), visual reasoning (Andreas et al., 2016; Johnson et al., 2017), visual storytelling (T.-H. Huang et al., 2016) and visual dialogue (De Vries et al., 2017). Another branch of research even combines language and vision with sound (Aytar, Vondrick, and Torralba, 2017). Two of such multimodal areas where reasoning about quantities is studied are computer vision and visual question answering.

In computer vision numerous models have been proposed for counting objects in an image (Segui, Pujol, and Vitria, 2015; C. Zhang et al., 2015; Arteta, Lempitsky, and Zisserman, 2016; J. Zhang et al., 2015; Chattopadhyay et al., 2017). One study (Stoianov and Zorzi, 2012) focused on modelling the Approximate Number Sense (ANS), the ability to gauge quantities without counting, and showed that it appears as a statistical property of images in generative neural networks. Relatively little attention has been given to the task of learning to assign an appropriate non-cardinal quantifier to a scene. The work of I. Sorodoc et al. (2016) was the first attempt to learn quantifiers 'none', 'some' and 'all' from visual data. Pezzelle et al. (Pezzelle, Marelli, and Bernardi, 2017) showed that learning quantifiers requires a vague estimation of the number of target objects in a scene, while learning cardinals makes use of provided information on the exact number of target objects. In (Pezzelle, I.-T. Sorodoc, and Bernardi, 2018) Pezzelle et al. presented a model that acquires a wider range of quantification mechanisms: set comparison, vague quantification and proportional estimation. (G. Chen, Deemter, and Lin, 2019) proposed a method for generating quantified descriptions of simple visual scenes.

The issue of quantification is particularly relevant in VQA where it naturally emerges in 'how many' questions. Numerical counting remains one of the biggest challenges in VQA (Chattopadhyay et al., 2017). Two typical approaches are inferring the count of objects from image embeddings learned by a CNN network (Kafle and Kanan, 2016; Kafle and Kanan, 2017b; Kafle and Kanan, 2017a) and identifying areas of the image containing relevant objects (Trott, Xiong, and Socher, 2017; Y. Zhang, Hare, and Prügel-Bennett, 2018). Relational networks that learn relationships between fragments of the image have also been used for counting (Santoro et al., 2017; Acharya, Kafle, and Kanan, 2019). Recently Nguyen et al. (Nguyen, Goswami, and X. Chen, 2020) presented a counting module that combines the query and the image locally and demonstrates state-of-the-art performance on visual counting benchmarks.

2.3 Learning quantitative reasoning on textual data

As has been noted previously, we formulate our problem as a textual inference problem. Although a few variants of inference tasks that combine textual and visual modalities have been proposed (Xie et al., 2019; Vu et al., 2018; Lai, 2018), we will use the NLI approach, where both the premise and the hypothesis are given in textual form.

The introduction of large datasets (Cooper et al., 1996; Marelli et al., 2014; S. R. Bowman et al., 2015; Williams, Nangia, and S. R. Bowman, 2017) for NLI inspired the development of multiple neural network models for this task (A. P. Parikh et al., 2016; Nie and Bansal, 2017; Conneau et al., 2017; Balazs et al., 2017; Q. Chen et al., 2016; Radford et al., 2018; Devlin et al., 2018). The importance of distinguishing quantitative reasoning as a subproblem that needs to be separately explored in NLI has been highlighted in (Sammons, Vydiswaran, and Roth, 2010; Clark, 2018; Bentivogli et al., 2010). It has been estimated that 4% of errors made by state-of-the-art models are caused by their insufficient understanding of numbers (Naik et al., 2018). (De Marneffe, Rafferty, and Manning, 2008) argues that 29% of contradictions in a corpus consisting of real-life contradictory pairs emerge from numeric discrepancies. Quantification is a complex functional semantic phenomenon that is a focus area in formal literature on NLI (Icard III and Moss, 2014). The only dataset obtained so far to evaluate models on quantitative inferences is EQUATE (Evaluating Quantitative Understanding Aptitude in Textual Entailment) (Ravichander et al., 2019).

A similar numerical task is arithmetic word problems (Hosseini et al., 2014; Mitra and Baral, 2016; Zhou, Dai, and L. Chen, 2015; Upadhyay et al., 2016; D. Huang et al., 2017; Ling et al., 2017), although it requires substantially less lexical proficiency because of the restricted nature of the text (Hosseini et al., 2014). A linguistic phenomenon closely related to quantifiers that has been explored in several studies in NLI is monotonicity (Mineshima et al., 2015; Abzianidze, 2015; H. Hu et al., 2019; Yanaka, Mineshima, Bekki, Inui, Sekine, et al., 2019a; Yanaka, Mineshima, Bekki, Inui, Sekine, et al., 2019b; Richardson et al., 2020; Yanaka, Mineshima, Bekki, and Inui, 2020). Quantifiers have two monotonicity directions: downward or upward entailment (Barwise and Cooper, 1981; Ladusaw, 1980; Van der Wouden, 2002). Upward entailing quantifiers such as 'more than', 'at least' and 'some' allow to make inferences about supersets of the set in the sentence that acts as a premise, while downward entailing quantifiers like 'less than', 'at most' and 'no' entail statements about subsets. The issue of monotonicity will not be explored in this project since we will work with premises that contain information about the exact number of objects of different types in a scene.

Numerical reasoning has been given particular attention in NLU tasks. Ria et al. proposed a framework that infers quantities from a given sentence (Roy, Vieira, and Roth, 2015). Wallace et al. (Wallace et al., 2019) showed that the NAQANet question answering model by Dua et al. (Dua et al., 2019) demonstrates high performance when answering questions that require numerical reasoning and that pre-trained token embeddings capture the value of numbers. Geva et al. (Geva, Gupta, and Berant, 2020) suggested pre-training BERT on synthetic numerical and textual data to make the model learn numerical reasoning as the first stage of the training process and then fine-tuning the model on a quantitative reasoning dataset. (Andor et al., 2019) uses a set of executable programs to perform numerical operations. (Ran et al., 2019) proposed a numerically-aware graph neural network to answer questions that require numerical reasoning. (K. Chen et al., 2020) presented a model that combines an attention neural network with context graph analysis for numerical reasoning over text.

Generalized quantifiers, however, are a relatively unexplored topic in NLI. Most models built for processing inferences with quantifiers use a logical approach with automated theorem proving (Tian, Miyao, and Matsuzaki, 2014; Dong, Tian, and Miyao, 2014; Mineshima et al., 2015; Abzianidze, 2016; Haruta, Mineshima, and Bekki, 2020). Neural approaches for quantitative reasoning in NLI have not been sufficiently investigated, which serves as motivation for our project. Bowman et al. (S. Bowman, Potts, and Manning, 2015) showed that tree-structured recursive neural network models perform well at learning to quantify on sentences generated based on a simple artificial grammar. Some shortcomings of their work were identified in (Veldhoen and Zuidema, 2017; Mul, 2018). Similar to our work is the study (Geiger et al., 2018), in which several neural network architectures are tested on artificially generated NLI examples with quantifiers, modifiers and negation.

Analysis of previous research related to our project suggests that quantified NLI is an essential step to modelling human intelligence and few studies have focused on this task, achieving limited progress. Some weak points of current neural models used for NLI have been highlighted in the literature.

A shortcoming of standard neural network models is that they often rely on shallow heuristics instead of learning the underlying general principles (J. Wang et al., 2017; Agrawal, Batra, and D. Parikh, 2016). It has been shown that NLI models adopt syntactic heuristics, like looking for lexical cues, such as correspondence of words (McCoy, Pavlick, and Linzen, 2019). (Yanaka, Mineshima, Bekki, and Inui, 2020) note that if the vocabulary of the training set remains unchanged in the test set but the syntactic structures are modified in relation to the training set, then models lose their generalization ability. (Veldhoen and Zuidema, 2017) investigate the findings of the study by Bowman et al. (S. Bowman, Potts, and Manning, 2015) and argue that the experiments performed in the paper do not prove that the models have learned the meaning of quantifiers, in fact, the models fail a more rigorous test.

Ravichandler et al. outlined several challenges for quantitative reasoning models for NLI:

complex fusion of verbal and quantitaive comprehension, understanding of wide range of lexical phenomena, such as hypernymy and hyponymy, and set comparison (Ravichander et al., 2019).

Chapter 3

Methodology

3.1 Quantified NLI for VI

Our goal is to verify how well the model can distinguish quantified statements that are correct given a complete description of a visual scene from quantified statements that are not. We view the sentence containing the description of an image as a premise and the quantified statement as a hypothesis in an inference relationship. Therefore our task is a classification task, where for every premise-hypothesis pair we predict one of two classes: 1 (hypothesis follows from the premise) and 0 (hypothesis does not follow from the premise).

The task **Quantified NLI for VI** can be defined more formally as follows: for a description of a visual scene P and a quantified statement H we determine if H follows from P.

The second class includes examples for which we can definitely say that the hypothesis is false given the premise, but also examples for which we cannot say if the hypothesis is true or false given the premise. An example for the latter would be the premise "there are 3 blue circles, there are 2 red triangles, there are 5 blue triangles" and the hypothesis "3 squares are red". As there are no squares of any kind in the premise, we cannot judge if any statement about squares is true or false given the information provided in the premise.

3.2 Framework for answering the research question

Framework. We conduct experiments that allow us to verify how well a neural network can learn to evaluate quantities based on provided complete descriptions of geometrical scenes.

In each stage of our experiments we explore a different group of quantifiers by following the same process:

1. Generate training and test data
2. Evaluate models on this data
3. Analyse the results of the experiments

Data. To answer the research question, we generate data for our experiments in the following way:

The dataset consists of premise-hypothesis pairs with a class label.

For simplicity we consider premises of the same structure that consist of 3 short statements of the form: 'there are'/'there is' + quantifier + color + type of object. Example: 'there are 4 red circles, there are 5 blue squares, there are 3 yellow circles'. We restrict the scope of our study to small domain size 3-27 keeping numbers in the premises between 1 and 9. Also for simplicity we consider only 3 colors: red, blue and yellow, and 3 types of objects, or shapes: circles, squares and triangles.

We generate a set of correct hypotheses and a set of incorrect hypotheses for every premise. We distinguish single attributes (colors or shapes) and attribute pairs (color-shape combinations). To obtain hypotheses we record the number of objects with single attributes and we record the number of objects with an attribute pair in the premise. Based on this information from the premise, we generate correct and incorrect hypotheses. The number of correct and the number of incorrect hypotheses per premise remain constant for each premise.

Hypotheses are of two types:

1. Hypotheses with color-shape attribute pairs of the form: quantifier + shape + color. Example: '3 circles are yellow'

2. Hypotheses with either colors or shapes of the form: quantifier + color/shape. Example: '7 objects are circles'

We obtain a dataset that consists of premises with the corresponding hypotheses and class label (0 or 1) by sampling the chosen number of correct hypotheses from the list of correct hypotheses and the chosen number of incorrect hypotheses from the list of incorrect hypotheses.

Description of experiments. For each group of quantifiers we first generate a dataset, then we optimize hyperparameters of the model and test the model with the selected hyperparameters on the test set. As a second experiment we investigate how the amount of negative and positive examples in the data influences the quality of prediction. Then we perform generalization experiments.

In order to test the model's ability to generalize, we designed experiments where we test the model on data that is different in some aspect from the training data. Our main goal is to see how well the model learns to understand quantifiers and their logical properties. Changing some parameter of the data in the test set allows us to gauge what role this parameter plays in the model's prediction.

Understanding quantifiers implies being able to tell what quantifier is correct for a visual scene with any objects of any colors or shapes. This motivates the first generalization experiment, where for testing we generate a dataset with attributes that do not appear in the training set, so the model is required to predict the class of examples with attributes it has not seen during training. The new attributes are 3 new colors and 3 new shapes instead of the ones in the training data: green, black, orange, spheres, cones, cubes. We train on data with old colors and shapes (blue, red, yellow, circles, triangles, squares) and test on data with new colors and shapes.

Another important aspect is how the model handles different domain sizes, or the total number of objects in the scene. In other words, the domain size is the sum of numbers in the three parts of the premise. When we change the domain size of the premise, we change the combinations of numbers in the premise. Smaller domain size means the numbers in the premise will tend to be smaller and bigger domain size implies that the numbers in the premise will tend to be bigger. Naturally the model will get accustomed to seeing certain numbers combined with color-shape combinations and predicting if a quantified statement is true for this kind of premises. To verify how the model handles data where numbers that appear most frequently are not seen as frequently during training, we train on examples that have a different domain size from examples in the test set. The numbers in premises are 1-9, so possible domain size is 3-27. We split the array of possible domain size in the middle at 15 and put all examples with domain size $<= 15$ in one set and all examples with domain size > 15 in another set and use these sets as a training set and a test set respectively in the case a) and as a test set and training set in the case b). In order to reason about the difference in prediction accuracy between the two cases, we build plots for the number of occurrences of numbers in the examples in the training set and the test set.

Having tried replacing the whole set of attributes, we also want to maintain the same set of attributes and see how the model handles varying combinations of colors and shapes. We train on data with one set of color-shape combinations and test on data with another set of color-shape combinations. We make sure that all attributes are present both in the training set and in the test set. We train on data with a bigger set of attribute combinations and test on a smaller set of attribute combinations that is unseen during training. We train on data with 6 color-shape combinations and test on data with 3 color-shape combinations. We randomly pick 3 combinations that contain all colors and all shapes and generate a dataset with these combinations to use as a test set. The training set has 6 remaining combinations.

In the last generalization experiment we verify how well the model classifies examples with a different form of premises. We change premises in the test set to a more natural form instead of an enumeration of the same type of short statements. We train on data with premises of the form 'there are n (color) (shape), there are m (color) (shape), there are k (color) (shape)' (that we considered previously) and testing on data with premises of the form 'there are n (color) (shape), m (color) (shape) and k (color) (shape)'. The reasoning here is that in the first type of premises numbers and attributes are always in a fixed position (3 last words in each of the 3 parts of the premise) and in the second type this is not the case.

First, we present the results of all generalization experiments, then for each experiment individually we provide a brief description of the training set and the test set and perform analysis of the results looking at one run of the experiment.

3.3 Model architecture

For our experiments we chose an LSTM based network architecture. LSTM networks are frequently used in Natural Language Processing.

LSTM (Long Short-Term Memory) networks (Hochreiter and Schmidhuber, 1997) belong to the class of recurrent neural networks (RNNs). RNNs have long been the preferred neural network architecture in cases where the data is sequential, like text data. This is because RNNs use the information about the previous inputs while processing the current input. RNNs do that through loops that pass on the result of processing the previous step, called a hidden state, to the current step. The scheme of an RNN can be seen in Figure 3.1.

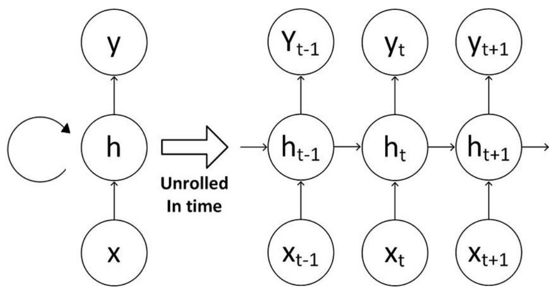


Figure 3.1: Recurrent neural network (Varsamopoulos, Bertels, and Almudever, 2019)

RNNs have several advantages: they can be used with inputs of any length, the increase of input size does not influence the model size, the information from previous timesteps is preserved. RNNs have found numerous applications: speech recognition, image captioning, translation, language modeling and others. However, RNNs have some shortcomings. In back propagation long-term gradients can either approach zero, which stops the learning of the model, or go to infinity, which makes the model unstable. LSTMs were created as a way to overcome these limitations. LSTMs use gates to control the information flow in the network. The structural unit of a LSTM network is a LSTM cell. A LSTM cell has a forget gate, an input gate and an output gate. The output of a LSTM network is the last hidden state which is a vector representation of the sentence that carries information about all parts of the sequence.

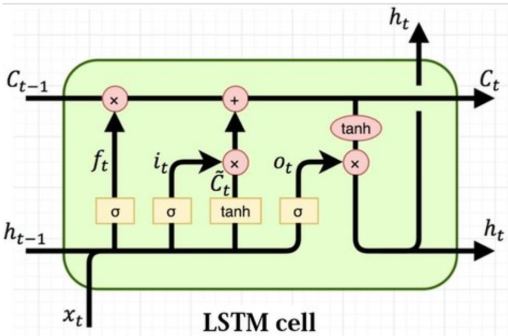


Figure 3.2: LSTM cell (Varsamopoulos, Bertels, and Almudever, 2019)

The equations of a LSTM cell for a time step $t = 1, ..., T$:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = \sigma(f_t \odot c_{t-1} + i_t \odot \tilde{c_t})$$

$$h_t = o_t \odot \tanh(c_t)$$

where x_t is the input vector of the cell, f_t is the forget gate, i_t is the input gate, o_t is the output gate, h_t is the hidden state vector, $\tilde{c_t}$ is the cell input, c_t is the cell state, W and U are weight matrices, b is the bias vector, σ is the sigmoid activation function, $\odot$ denotes element-wise product.

LSTM networks have been successfully used in NLI. Wang et al. proposed a modified LSTM network for sequential matching of the premise and the hypothesis (S. Wang and Jiang, 2015). Zhang et al. presented a model that uses sentence fusion modules on representations of the premise and the hypothesis generated by the LSTM encoder (S. Zhang, S. Liu, and M. Liu, 2017). Chen et al. (Q. Chen et al., 2016) argue that LSTM networks often perform better than more complex models and that they have unexplored potential for NLI.

The choice of architecture was inspired by the paper by Bowman et al. that introduced the SNLI corpus (S. R. Bowman et al., 2015) and the papers (B. Hu et al., 2014) and (Mou et al., 2015). This is a so-called siamese architecture where vector representations of two sentences get concatenated and then passed to the fully connected layers. Mueller et al. successfully used a siamese LSTM architecture to evaluate semantic similarity between sentences (Mueller and Thyagarajan, 2016). In (S. R. Bowman et al., 2015) Bowman et al. build a model with an LSTM layer that encodes the premise and the hypothesis separately and several fully connected layers. We implemented a very simple network with an LSTM layer, one fully connected hidden layer and a fully connected output layer. The architecture of our neural network model is illustrated in Figure 3.3.

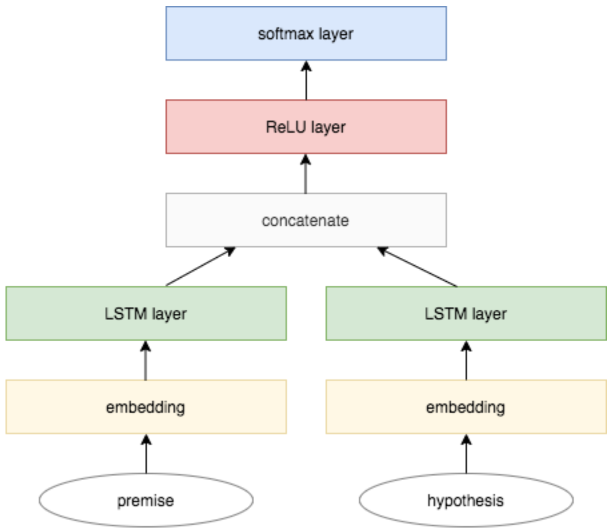


Figure 3.3: Model architecture

3.4 Quantifiers covered in this project

In this thesis we explore the three following groups of quantifiers.

1. **Numerical quantifiers.** This type of quantifiers require counting, including associating 0 with absent attributes. The operations needed to correctly determine the truth value of a hypothesis given a premise are addition of numbers from different parts of the premise, if the attribute in the hypothesis appears in several parts of the premise, or just matching the information in one part of the premise with the information in the hypothesis.

2. **Quantifiers 'no', 'some', 'all'.** These quantifiers were the subject of the visual quantification study by Sorodoc et al. (I. Sorodoc et al., 2016). As was mentioned above, these quantifiers are similar in terms of semantic complexity to numerical quantifiers in that they can be computed by finite state automata. This type of quantifiers require learning that 'no' should be associated with absent attributes, 'some' should be associated with attributes present in the visual scene no matter the number in the premise and 'all' should be associated with the sizes of the sets A, B in all(A, B) being equal. To check if a hypothesis with quantifiers 'no', 'some', 'all' is true given a premise, it is necessary, similar to the

previous case, to consider information in a single part of the premise (for 'some' and 'no) or possibly combine information from different parts of the premise (for 'all').

3. **Quantifiers 'more than n%', n=0,...,99, 'less than n%', n=1,...,100, and 'exactly n%', n=0,...,100.** The truth value of this type of quantifiers cannot be computed by automata without using memory. These are more advanced quantifiers that require matching percentage values in the hypothesis with numbers in the premise for 'exactly $n\%$' and also comparison for quantifiers 'more than $n\%$' and 'less than $n\%$'. For all of these quantifiers to check if they apply to a visual scene, like numerical quantifiers, it is either necessary to consider just one part of the premise or to combine information from different parts of the premise.

We cover a wide range of quantifiers that would allow us to check different aspects of quantitative reasoning ability acquired by a machine learning model. Vague quantifiers like 'many' were not considered because of difficulty of interpretation.

3.5 Applying the framework to different quantifiers

3.5.1 Numerical quantifiers

3.5.1.1 Data generation

Premises generation

There are 3 colors and 3 shapes, therefore there are in total 9 color-shape combinations. There are 81 possible short descriptions with numbers 1-9, 3 colors and 3 shapes. Since premises consist of 3 short descriptions and the combinations of colors and shapes should all be different in a premise, there are $81 * 72 * 63/3! = 61236$ possible premises. For every premise out of this list of premises we generate a set of correct and incorrect hypotheses.

Hypotheses generation

For every item in the list of all attributes present in the premise:

1. We generate correct hypotheses for single attributes and attribute pairs and add them to a list of correct hypotheses for the premise.

2. We add a hypothesis with 0 to the set of incorrect hypotheses.

3. We generate incorrect hypotheses by randomly selecting a number that is different from the number in the premise, an incorrect number, and add them to a list of incorrect hypotheses for the premise. For single attributes the selected number can be from 1 to 27, for attribute pairs the selected number can be from 1 to 9 because of how we build premises.

For every item in the list of all attributes not present in the premise:

4. We generate incorrect hypotheses for attribute pairs by going through the list of attributes not present in the premise and randomly selecting an incorrect number and add the generated hypotheses to the list of incorrect hypotheses for the premise. If the shape does not appear in the premise, then no statement with that shape follows from the premise, so the hypotheses with this shape as a single attribute or with this shape as one of the attributes in the attribute pair are all incorrect. Therefore in this case we select a random number

out of the set 0-9. If the shape appears in the premise, then we add a hypothesis with 0 to the set of correct hypotheses and generate incorrect hypotheses randomly selecting a number out of the set 1-9 and add them to the set of incorrect hypotheses for the premise.

5. Because we can always make statements about what number of objects have a certain shape or color, for single attributes not present in the premise we add a hypothesis with 0 to the set of correct hypotheses and we add an incorrect hypothesis randomly selecting a number out of the set 1-27 and add them to the set of incorrect hypotheses for the premise.

To summarize, the types of hypotheses are:
Correct hypotheses types:

1. Hypotheses with single attributes present in the premise
2. Hypotheses with attribute pairs present in the premise
3. 0-hypotheses with single attributes not present in the premise
4. 0-hypotheses with attribute pairs not present in the premise when the shape is in the premise

Incorrect hypotheses types:

5. Hypotheses with single attributes present in the premise
6. Hypotheses with attribute pairs present in the premise
7. Hypotheses with single attributes not present in the premise
8. Hypotheses with attribute pairs not present in the premise
9. 0-hypotheses with single attributes present in the premise
10. 0-hypotheses with attribute pairs present in the premise
11. 0-hypotheses with attribute pairs not present in the premise when the shape is not in the premise

The distribution of hypotheses among these 11 types can be seen in Figure 3.4.

In this part of experiments by design of our data there are a few principles that lead to a correct prediction:

1. If the hypothesis is about single attributes, all numbers next to all occurrences of this attribute in three parts of the premise should be added up.
2. If the hypothesis pertains to a pair of attributes, then the number in the part of the premise that contains this attribute pair if it is present in the premise should be equal to the number in the hypothesis. In other words, only one part of the premise should be considered when judging if a hypothesis with an attribute pair follows from the premise.
3. If an attribute or an attribute pair is not in the premise, a hypothesis containing them with 0 is generally correct (except for the case when the shape in the color-shape pair is not in the premise); the reverse is also true.

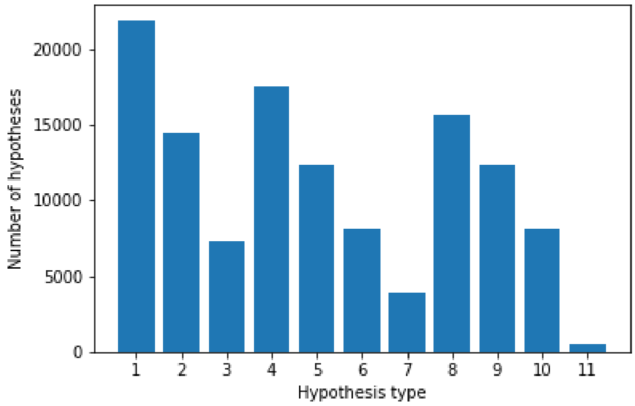


Figure 3.4: Number of hypotheses of each type in the dataset

3.5.1.2 Research subquestions

In the first experiment we set out to answer the following subquestion:

RQ 1.1: based on a complete description of a visual scene containing geometrical objects, is a simple LSTM-based neural network model able to learn to infer which statements containing numerical quantifiers about the objects in the scene are true and which are false?

The second experiment allows to answer the following subquestion:

RQ 1.2: how does the quality of prediction of the model change with the decrease or increase of positive/negative examples in the data with numerical quantifiers?

For the generalization experiments we formulate the following subquestions:

RQ 1.3: does the model generalize well to data with numerical quantifiers where attributes are unseen during training?

RQ 1.4: does the model generalize well to data with numerical quantifiers where domain size is unseen during training?

RQ 1.5: does the model generalize well to data with numerical quantifiers where attribute combinations are unseen during training?

RQ 1.6: does the model generalize well to data with numerical quantifiers where the form of premises is unseen during training?

3.5.2 Quantifiers 'no', 'some', 'all'

3.5.2.1 Data generation

Premises remain the same as in stage 1. Premises consist of 3 short statements of the form: 'there are'/'there is' + quantifier + color + shape.

Hypotheses generation

For every item in the list of single attributes and attribute pairs (type-color) that are present in a premise and their number of occurrences:

Case 1. The number of occurrences of a single attribute or an attribute pair in a premise is equal to the total number of objects in that premise: we add to the set of correct hypotheses a statement with the quantifier 'all', we also add to the set of correct hypotheses a statement with the quantifier 'some' and we add to the set of incorrect hypotheses a statement with the quantifier 'no'. We consider any case where 'all' is applicable a case where 'some' is applicable too.

Case 2. The number of occurrences of a single attribute or an attribute pair is less than the total number of objects: we add to the set of correct hypotheses a statement with the quantifier 'some' and we add to the set of incorrect hypotheses a statement with the quantifier 'no' and a statement with the quantifier 'all'.

For every item in the list of single attributes and attribute pairs that do not appear in a premise:

We add to the set of correct hypotheses a statement with the quantifier 'no' and we add to the set of incorrect hypotheses a statement with the quantifier 'some' and a statement with the quantifier 'all'.

The logical relationship between the quantifiers 'no', 'some', 'all':

1. all $\implies$ some
2. all $\oplus$ no
3. some $\implies$ $\not$all
4. some $\oplus$ no

There are 8 types of hypotheses:

Correct hypotheses types:

1. Hypotheses with single attributes present in the premise
2. Hypotheses with attribute pairs present in the premise
3. Hypotheses with single attributes not present in the premise
4. Hypotheses with attribute pairs not present in the premise

Incorrect hypotheses types:

5. Hypotheses with single attributes present in the premise
6. Hypotheses with attribute pairs present in the premise
7. Hypotheses with single attributes not present in the premise
8. Hypotheses with attribute pairs not present in the premise

3.5.2.2 Research subquestions

Analogously to numerical quantifiers, in the first experiment we set out to answer the following subquestion:

RQ 2.1: based on a complete description of a visual scene containing geometrical objects, is a simple LSTM-based neural network model able to learn to infer which statements containing quantifiers 'no', 'some', 'all' about the objects in the scene are true and which are false?

The second experiment allows to answer the following subquestion:

RQ 2.2: how does the quality of prediction of the model change with the decrease or increase of positive/negative examples in the data with quantifiers 'no', 'some', 'all'?

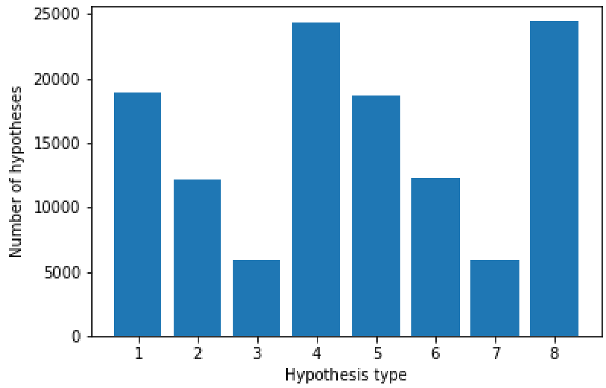


Figure 3.5: Number of hypotheses of each of the 8 types in the dataset

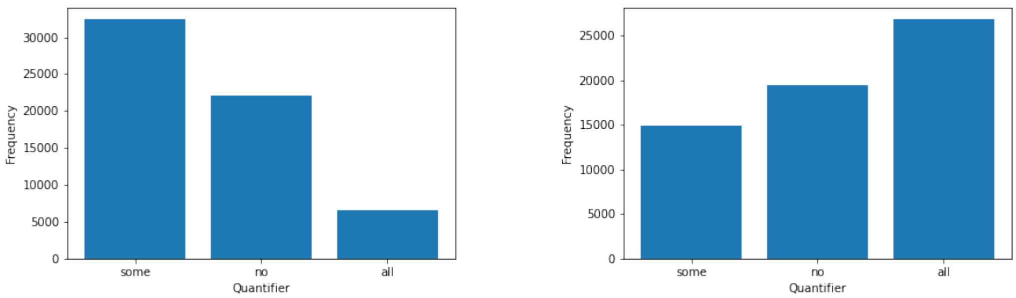


Figure 3.6: Distribution of quantifiers 'no', 'some', 'all' in correct and incorrect hypotheses of the generated dataset

For the generalization experiments we formulate the following subquestions:

RQ 2.3: does the model generalize well to data with quantifiers 'no', 'some', 'all' where attributes are unseen during training?

RQ 2.4: does the model generalize well to data with quantifiers 'no', 'some', 'all' where domain size is unseen during training?

RQ 2.5: does the model generalize well to data with quantifiers 'no', 'some', 'all' where attribute combinations are unseen during training?

RQ 2.6: does the model generalize well to data with quantifiers 'no', 'some', 'all' where the form of premises is unseen during training?

3.5.3 Quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$'

3.5.3.1 Data generation

Premises remain the same as in stage 1. Premises consist of 3 short statements of the form: 'there are'/'there is' + quantifier + color + shape.

Hypotheses generation

For attributes/attribute pairs present in the premise:

1. If the number of objects with an attribute/attribute pair is equal to a percentage k of objects of this shape for attribute pairs and of the total number of objects for single attributes, where k is integer, then we add the corresponding hypothesis with 'exactly $k\%$' to the set of correct hypotheses and a hypothesis with 'exactly $n\%$' with a randomly selected number $n = 1, ..., k-1, k+1, ..., 100$ that is not correct to the set of incorrect hypotheses. If the percentage k is not integer, we add a hypothesis with 'exactly $n\%$' with a randomly selected number $n = 1, ..., 100$ to the set of incorrect hypotheses. We also add a hypothesis with 'exactly 0%' to the set of incorrect hypotheses.

2. If the percentage (of objects of this shape for attribute pairs and of the total number of objects for single attributes) is $k\%$, 'less than $n\%$' is true for any $n = m, ..., 100, m = [k]+1$, if k is a real number, $m = k+1$, if k is an integer and 'less than $n\%$' is false for any $n = 1, ..., m, m = [k]$, if k is a real number, and $m = 1, ..., k$, if k is integer; 'more than $n\%$' is true for any $n = 0, ..., m, m = [k]$, if k is a real number, $m = k-1$, if k is an integer, and 'more than $n\%$' is false for any $n = m, ..., 99, m = [k]+1$, if k is a real number, $m = k$, if k is an integer.

For attributes/attribute pairs not present in the premise:

1. For attribute pairs that do not appear in the premise where the shape is in the premise we add a hypothesis with 'less than $n\%$' where n is randomly selected from the set $1, ..., 100$ and a hypothesis with 'exactly 0%' to the set of correct hypotheses; we add a hypothesis with 'more than $n\%$' where n is randomly selected from the set $0, ..., 99$ and a hypothesis with 'exactly $n\%$' where n is randomly selected from the set $1, ..., 100$ to the set of incorrect hypotheses. If the shape from the attribute pair is not in the premise, we add a hypothesis with 'less than $n\%$' where n is randomly selected from the set $1, ..., 100$, a hypothesis with 'more than $n\%$' where n is randomly selected from the set $0, ..., 99$ and a hypothesis with 'exactly $n\%$' where n is randomly selected from the set $0, ..., 100$ to the set of incorrect hypotheses.

2. For single attributes that do not appear in the premise where the shape is in the premise we add a hypothesis with 'less than $n\%$' where n is randomly selected from the set $1, ..., 100$ and a hypothesis with 'exactly 0%' to the set of correct hypotheses; we add a hypothesis with 'more than $n\%$' where n is randomly selected from the set $0, ..., 99$ and a hypothesis with 'exactly $n\%$' where n is randomly selected from the set $1, ..., 100$ to the set of incorrect hypotheses.

The logical relationship between quantifiers 'less than $n\%$' and 'more than $n\%$' and 'exactly $n\%$':

1. The attribute/attribute pair is not present in the premise $\Longrightarrow$ 'exactly 0%' is true and therefore 'less than $n\%$' is true for any $n = 1, ..., 100$.

   The attribute/attribute pair is not present in the premise $\Longrightarrow$ 'exactly 0%' is true and therefore 'more than $n\%$' is false for any $n = 0, ..., 99$.

2. The attribute/attribute pair is present in the premise and 'exactly 100%' is true $\Longrightarrow$ 'less than $n\%$' is false for any $n = 1, ..., 100$.

   The attribute/attribute pair is present in the premise and 'exactly 100%' is true $\Longrightarrow$ 'more than $n\%$' is true for any $n = 0, ..., 99$.

3. The attribute/attribute pair is present in the premise and 'exactly 100%' is false, so the correct percentage value $k\%$ is between 0 and 100: $0 < k < 100 \implies$ 'less than $n\%$' is true for any $n = m, ..., 100, m = [k] + 1$, if k is a real number, $m = k + 1$, if k is an integer and 'less than $n\%$' is false for any $n = 1, ..., m, m = [k]$, if k is a real number, and $m = 1, ..., k$, if k is integer.

   The attribute/attribute pair is present in the premise and 'exactly 100%' is false, so the correct percentage value $k\%$ is between 0 and 100: $0 < k < 100 \implies$ 'more than $n\%$' is true for any $n = 0, ..., m, m = [k]$, if k is a real number, $m = k - 1$, if k is an integer, and 'more than $n\%$' is false for any $n = m, ..., 99, m = [k] + 1$, if k is a real number, $m = k$, if k is an integer.

Therefore the hypotheses types are:
Correct hypotheses types:

1. Hypotheses with 'more than $n\%$' with single attributes present in the premise
2. Hypotheses with 'more than $n\%$' with attribute pairs present in the premise
3. Hypotheses with 'less than $n\%$' with single attributes present in the premise
4. Hypotheses with 'less than $n\%$' with attribute pairs present in the premise
5. Hypotheses with 'exactly $n\%$' with single attributes present in the premise
6. Hypotheses with 'exactly $n\%$' with attribute pairs present in the premise
7. Hypotheses with 'less than $n\%$' with single attributes not present in the premise
8. Hypotheses with 'less than $n\%$' with attribute pairs not present in the premise
9. Hypotheses with 'exactly 0%' with single attributes not present in the premise
10. Hypotheses with 'exactly 0%' with attribute pairs not present in the premise when the shape is in the premise

Incorrect hypotheses types:

11. Hypotheses with 'more than $n\%$' with single attributes present in the premise
12. Hypotheses with 'more than $n\%$' with attribute pairs present in the premise
13. Hypotheses with 'less than $n\%$' with single attributes present in the premise
14. Hypotheses with 'less than $n\%$' with attribute pairs present in the premise
15. Hypotheses with 'exactly $n\%$' with single attributes present in the premise
16. Hypotheses with 'exactly $n\%$' with attribute pairs present in the premise
17. Hypotheses with 'exactly 0%' with single attributes present in the premise
18. Hypotheses with 'exactly 0%' with attribute pairs present in the premise
19. Hypotheses with 'more than $n\%$' with single attributes not present in the premise
20. Hypotheses with 'more than $n\%$' with attribute pairs not present in the premise

21. Hypotheses with 'less than $n\%$' with attribute pairs not present in the premise when the shape is not in the premise
22. Hypotheses with 'exactly $n\%$' with single attributes not present in the premise
23. Hypotheses with 'exactly $n\%$' with attribute pairs not present in the premise

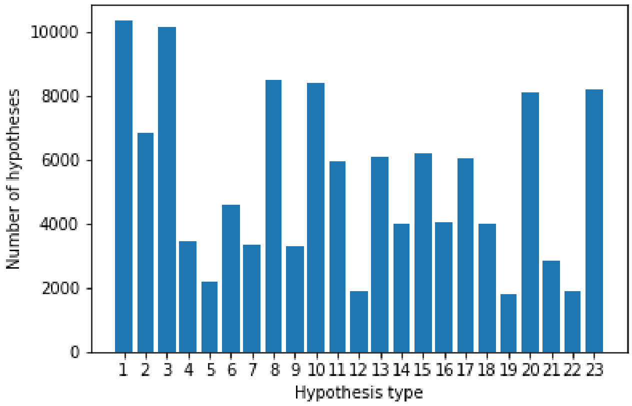


Figure 3.7: Number of hypotheses of each type in the dataset

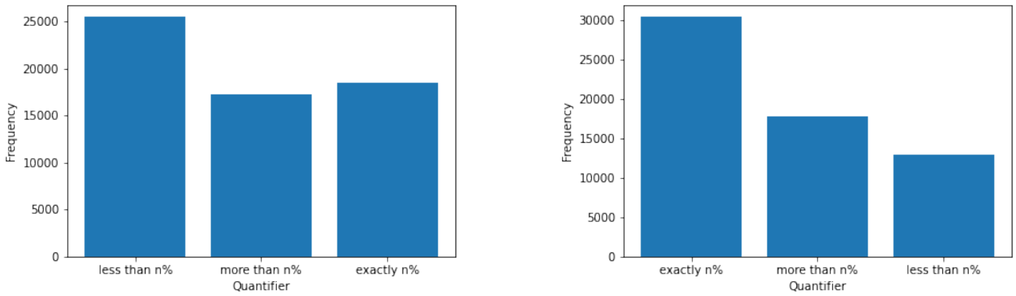


Figure 3.8: Distribution of quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' in correct and incorrect hypotheses of the generated dataset

3.5.3.2 Research subquestions

Analogously to previous parts of experiments, we define the following subquestions:

RQ 3.1: based on a complete description of a visual scene containing geometrical objects, is a simple LSTM-based neural network model able to learn to infer which statements containing quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' about the objects in the scene are true and which are false?

RQ 3.2: how does the quality of prediction of the model change with the decrease or increase of positive/negative examples in the data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$'?

RQ 3.3: does the model generalize well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where attributes are unseen during training?

RQ 3.4: does the model generalize well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where domain size is unseen during training?

RQ 3.5: does the model generalize well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where attribute combinations are unseen during training?

RQ 3.6: does the model generalize well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where the form of premises is unseen during training?

Chapter 4

Experiments

4.1 Experiment settings

The network consists of an input layer, a trainable embedding layer that creates 100 dimensional embeddings, an LSTM layer that encodes the premise and the hypothesis separately, a fully connected layer with ReLU activation and a fully connected softmax output layer. We use a Keras embedding layer that turns indexes from the vocabulary into vectors, so the embeddings do not capture the context of words. Gaussian noise with standard deviation 0.1 is added to LSTM encodings of the premise and the hypothesis, they are concatenated together with the result of their element-wise multiplication and subtraction and passed to the dense layer. The last layer is a 2-class softmax classifier. We did not experiment with different types of word embeddings as we did not set out to obtain the best performing model. We have tried several regularization methods to improve generalization of the network and the best performing one was adding Gaussian noise to LSTM encodings of the premise and the hypothesis. Since the goal of our study was to see how well a simple neural network learns to understand quantities, we did not set out to build the best performing model and therefore we did not experiment with the architecture by varying the number of layers and other parameters of the network.

In each stage of experiments the network hyperparameters (the number of nodes in the LSTM layer, the number of nodes in the fully connected layer and the learning rate) are optimized using Bayesian optimization with training set of size 64297, validation set of size 27557, test set of size 30618.

We use accuracy, precision, recall and F1 score as performance measures. The performance measures are averaged over 10 runs of the same experiment.

4.2 Numerical quantifiers

4.2.1 Main experiment

The best hyperparameter values found as a result of optimization are: 100 LSTM units, 200 dense units, learning rate: 0.00348541342823731.

The network with the selected values of hyperparameters was tested on the testing set:

Error analysis. As we can see recall is higher than precision, so most of the mistakes are due to false positives. 77.6% of incorrect predictions are false positives (857 out of 1104).

The plot in Figure 4.1 shows how many hypotheses that are incorrect but were predicted to

Accuracy	Precision	Recall	F1 score
96.39	94.57	98.37	96.43

Table 4.1: Performance measures for the model with the best hyperparameter values

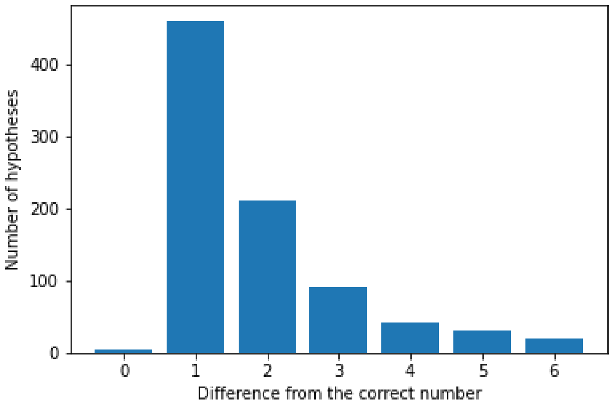


Figure 4.1: Difference between numbers in the hypotheses labelled true incorrectly and correct numbers

be correct have a certain value of difference between the incorrect number in the hypothesis and the correct number. It is evident that the model struggles with numbers close to the correct one in its predictions: the amount of incorrectly classified examples steadily decreases as the difference increases. Most examples that led to false positive predictions had difference of 1 from the correct number.

We also build a plot for accuracy for different value of difference between numbers in the incorrect hypotheses and correct numbers. This plot can be seen in Figure 4.2.

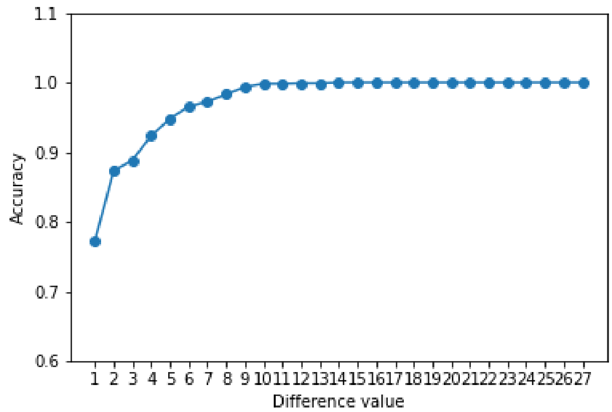


Figure 4.2: Accuracy for different values of difference between numbers in the incorrect hypotheses and correct numbers

Looking at the plot, we can conclude that the model does not handle well small discrepancies in the false examples (tends to label more such negative examples as true), but learns to predict well for cases when the difference between the true number and the incorrect number is substantial.

58% of false positives are with attribute pairs and 42% are with single attributes. It looks like the model struggles more with assigning an appropriate number to a color-shape combination which requires only repeating information from one of the three parts of the premise and deals better with reasoning about single attributes, color or shapes, although this sometimes requires to accumulate information from several parts of the premise. This might be explained by the fact that hypotheses with incorrect numbers with attribute pairs not present in the premise are more frequent in the dataset than hypotheses with attribute pairs present in the premise. We build the plot in Figure 4.3 to see how many incorrect predictions occur for examples where it is necessary to combine information about colors/shapes from different parts of the premise. It seems that again the model struggles slightly more when handling information in one part of the premise than in two parts of the premise, although, interestingly enough, there are almost no examples where a single attribute appeared in all three parts of the premise that were classified as positive incorrectly.

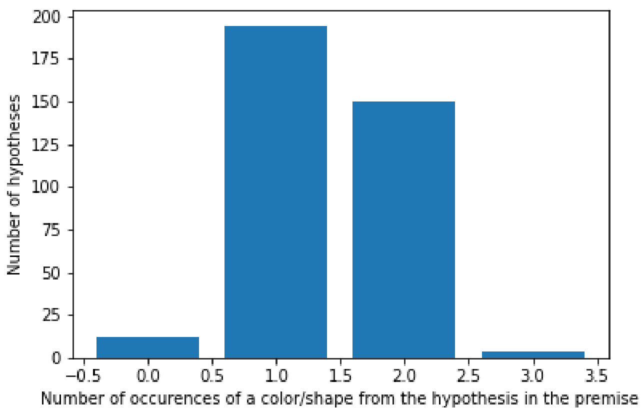


Figure 4.3: Number of occurrences of the attribute from the hypothesis in the premise for hypotheses of type quantifier + color/shape

By looking at the results, we can answer **research subquestion 1.1**: based on a complete description of a visual scene, our model can indeed learn to infer which statements with numerical quantifiers are true and which are false.

4.2.2 Exploring the influence of the amount of positive and negative examples

In the second experiment we try to see how the amount of positive and negative examples in the training data influences the quality of prediction. We train on data with different values of the number of correct and incorrect hypotheses per premise to see how it affects quality of prediction. We generate datasets for different values of the number of correct and incorrect hypotheses per premise: 122472 examples generated in total with 1 correct and 1 incorrect hypothesis per premise; 183708 examples with 1 correct and 2 incorrect hypotheses per premise;

the same number of examples with 2 correct and 1 incorrect hypotheses per premise; 244944 examples with 1 correct and 3 incorrect hypotheses per premise; the same number of examples with 3 correct and 1 incorrect hypotheses per premise.

In all cases data with 1 correct and 1 incorrect hypothesis per premise was used as a test set. In each case 61236 examples were sampled from the dataset with certain value of the number of correct and incorrect hypotheses per premise and were used as a training set and 61236 examples were sampled from the dataset with 1 correct and 1 incorrect hypothesis per premise and used as a test set.

Num of cor. and incor. hyp. per pr.	Accuracy	Precision	Recall	F1 score
3,1	74.42 ± 3.01	66.72 ± 2.88	98.01 ± 0.43	79.36 ± 1.98
2,1	82.33 ± 5.71	75.77 ± 6.69	96.63 ± 1.70	84.76 ± 4.17
1,1	87.79 ± 4.54	85.41 ± 5.08	91.34 ± 3.64	88.25 ± 4.26
1,2	87.57 ± 3.82	90.48 ± 2.75	83.87 ± 5.67	87.01 ± 4.22
1,3	81.46 ± 7.28	91.38 ± 1.89	69.22 ± 15.13	77.81 ± 11.72

Table 4.2: Results of experiment 2

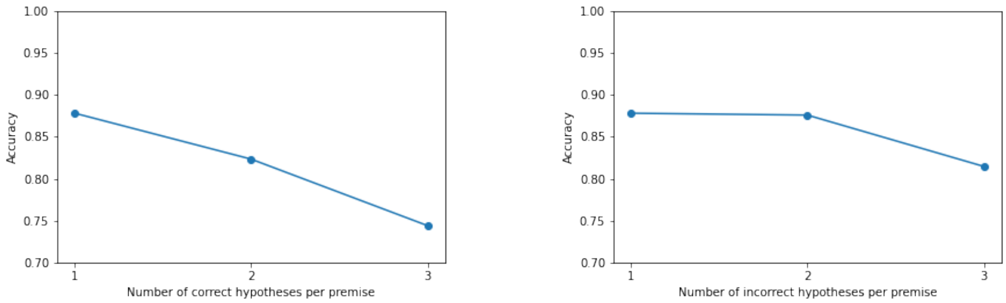


Figure 4.4: Accuracy depending on the number of correct and incorrect hypotheses per premise

Answering **research subquestion 1.2**, accuracy peaks when the number of both correct and incorrect hypotheses per premise in the data is 1 and then drops with the increase of the amount of correct and incorrect hypotheses in the data. Increasing the number of incorrect hypotheses that the model sees during training makes the data biased towards negative examples which means that the model is not exposed to enough of positive examples to be able to accurately predict one of two classes and the reverse is true for increasing the number of correct hypotheses. Recall significantly decreases with the increase of the number of negative examples and precision decreases with the increase of the number of positive examples.

4.2.3 Generalization experiments

1. **Attributes in the test set unseen during training.**

   As the training set 61236 randomly selected examples with old attributes were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', '8 objects are triangles', class: 0 (false). The model was tested on 61236 randomly selected examples with new attributes. An example

Num.	Aspect	Accuracy	Precision	Recall	F1 score
1	Attributes	68.21 ± 4.47	71.84 ± 2.96	60.03 ± 12.42	64.78 ± 7.45
2a	Domain size	85.07 ± 2.71	86.84 ± 3.36	82.78 ± 2.70	84.73 ± 2.69
2b	Domain size	80.94 ± 3.83	85.07 ± 4.78	75.25 ± 4.40	79.79 ± 4.01
3	Attribute combinations	60.97 ± 1.44	63.97 ± 2.14	50.48 ± 3.78	56.33 ± 2.41
4	Form of premises	86.16 ± 3.47	85.30 ± 4.24	87.59 ± 3.55	86.38 ± 3.32

Table 4.3: Results of generalization experiments with numerical quantifiers

of a premise-hypothesis pair from the test set: 'there are 3 green cubes, there are 5 black cones, there are 6 green cones', '11 objects are cones', class: 1 (true).

Error analysis. We verify what leads to mistakes in predictions on data with unseen attributes on one run of the experiment.

Accuracy	Precision	Recall	F1 score
67.75	71.77	58.82	64.65

Table 4.4: Performance measures for a run of experiment 1

We can see that precision is higher than recall, which means that most mistakes are due to false negatives, they constitute 64% of the wrong predictions. Out of false negatives 76% (9568 out of 12644) of examples have hypotheses with 0. 63% of those examples with 0-hypotheses have 0-hypotheses with attribute pairs and 37% of those 0-hypotheses contain single attributes. This means that the model mostly struggles to learn that when a pair of a color and a shape does not appear together in one part of the premise, the example should be classified as positive, and to a lesser degree that when a color or a shape does not appear in the premise, the example with a hypothesis with 0 containing this color or shape should be classified as positive.

80% of false positives have non-0 hypotheses. In 64% of those examples hypotheses are about pairs of colors and shapes, so again the model deals worse with attribute pair hypotheses.

To summarise, in this experiment both false negative and false positive predictions occur for examples with hypotheses containing attribute pairs, which means that the model does not handle well reasoning about a single part of the premise containing new attributes and associating either 0 or an appropriate number to the premise.

Answering **research subquestion 1.3**, when predicting on data with numerical quantifiers with unseen attributes, accuracy drops significantly, but the model still demonstrates an ability to distinguish whether a hypothesis is true or false given a premise with moderate accuracy.

2. **Domain size in the test set unseen during training.**

   a) As the training set 50000 randomly selected examples with domain size $<= 15$ were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', '8 objects are triangles', class: 0 (false). The model was tested on 50000 randomly selected examples with domain size > 15. An example of a premise-hypothesis pair from the test set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', '16 objects are circles', class: 1 (true).

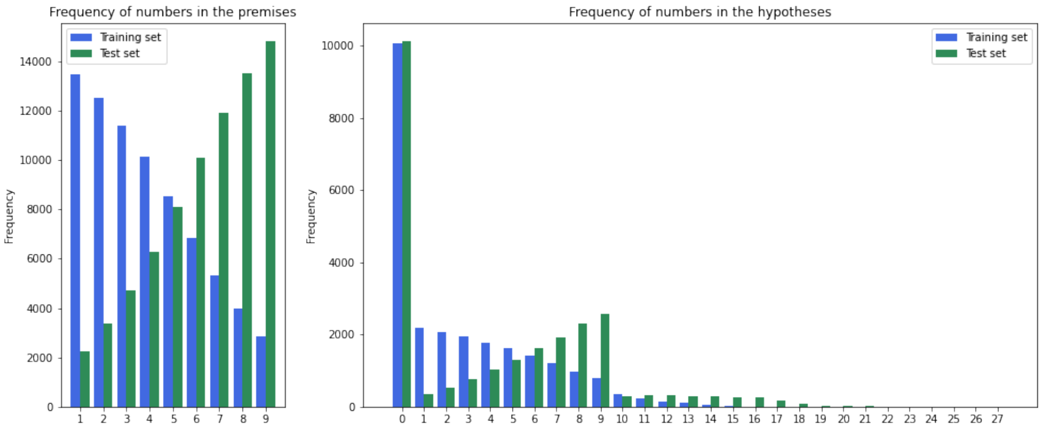


Figure 4.5: Frequency of numbers in premises and hypotheses in the training set and the test set for experiment 2a

Error analysis. We take one run of the experiment and explore the mistakes made in the predictions.

Accuracy	Precision	Recall	F1 score
86.31	89.04	82.83	85.82

Table 4.5: Performance measures for a run of experiment 2a

There are almost twice as many false negatives as false positives - 4297 versus 2550. 79% of hypotheses in examples wrongly classified as negative have numbers higher than 5. Out of the false positives 53% have hypotheses with 0 and 28% have hypotheses with numbers higher than 5. Those are values that the model saw less of during training as can be seen in Figure 4.5, so it is not able to make correct predictions for hypotheses with these type of numbers.

b) As the training set 50000 randomly selected examples with domain size > 15 were used. An example of a premise-hypothesis pair from the training set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', '16 objects are circles', class: 1 (true). The model was tested on 50000 randomly selected examples with domain size <= 15. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', '8 objects are triangles', class: 0 (false).

Error analysis.

Accuracy	Precision	Recall	F1 score
83.53	88.04	77.72	82.56

Table 4.6: Performance measures for a run of experiment 2b

Again there are almost twice as many false negatives as false positives. 71% of hypotheses in examples wrongly classified as negative have numbers lower than 5, but not 0. Out of the false positives 42% have hypotheses with 0 and 44% have hypotheses with numbers higher

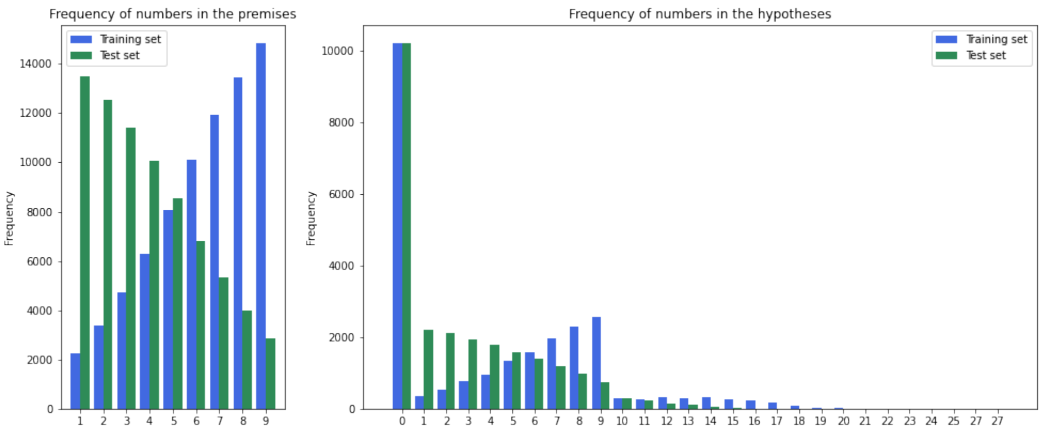


Figure 4.6: Frequency of numbers in premises and hypotheses in the training set and the test set for experiment 2b

than 5. The latter can likely be explained by the fact that hypotheses with larger numbers appeared in the positive examples during training, as in training we used examples with domain size higher than 15. Like in the previous case, incorrect predictions stem from the difference in values seen during training and values in test data.

Based on the results of both parts of the experiments, we can answer **research subquestion 1.4** and say that the model definitely generalizes well to data with domain size unseen during training with limited decrease in accuracy.

3. **Attribute combinations in the test set unseen during training.**

As the training set 29160 examples with old color-shape combinations were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue triangles, there are 8 red circles, there are 3 yellow squares', '8 objects are triangles', class: 0 (false). The model was tested on 1458 examples with new color-shape combinations. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', '8 objects are triangles', class: 0 (false).

Error analysis.

Accuracy	Precision	Recall	F1 score
62.35	67.24	48.15	56.12

Table 4.7: Performance measures for a run of experiment 3

There are more than twice as many false negatives as there are false positives. 52% of false negatives contain hypotheses with 0 and all of them are with attribute pairs. 48% of false negatives contain hypotheses with attribute pairs. This means that the model cannot reason about the combinations of colors and shapes it has not seen during training and will incorrectly classify the examples as negative because hypotheses with these color-shape combinations were likely in negative examples in the training set as those combinations were not present in the premise.

Results of the experiment allow us to answer **research subquestion 1.5**: the model is still able to infer what quantified statements with numerical quantifiers are true and which are false given a description of a scene, but with a substantial decrease in accuracy.

4. **Premises in the test set of the form unseen during training.**

   As the training set 61236 randomly selected examples with the old shape of premises were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', '8 objects are triangles', class: 0 (false). The model was tested on 61236 randomly selected examples with the new shape of premises. An example of a premise-hypothesis pair from the test set: 'there are 8 red squares, 8 red circles and 3 blue squares', '16 objects are red', class: 1 (true).

   Error analysis.

Accuracy	Precision	Recall	F1 score
84.79	83.63	86.51	85.05

   Table 4.8: Performance measures for a run of experiment 4

   Overall the model adapts quite well to the change in the form of premises if compared to the first part of experiment 2 with the initial dataset in a similar setting. Recall is higher than precision. 51% of examples wrongly classified as positive contain hypotheses with 0 and most of them are with attribute pairs present in the premise. 51% of examples wrongly classified as positive contain non-0 hypotheses with attribute pairs with either an incorrect number corresponding to an attribute pair present in the premise or an incorrect positive number corresponding to an attribute pair that is not in the premise.

   Answering **research subquestion 1.6**, we can definitely say that the model generalizes well to data with unseen type of premises.

4.2.4 Discussion

To summarize, the network successfully learns to correctly identify whether a quantified statement with a numerical quantifier is true or false given the complete description of a scene with accuracy of 96.4%, when the test data comes from the same distribution as the training data. Experiments show, however, that the network struggles to generalize beyond the data seen during training and the accuracy on data that differs from training data in some experiments drops quite significantly. The change of domain size between training set and test set only slightly affects the accuracy, as does the change in the form of the premises. The situation is very different when we change the attributes (attributes themselves like in experiment 1 or attribute combinations like in experiment 3). Predicting for color-shape combinations unseen during training was the hardest for the model. It seems like most mistakes stem from the model not being able to judge if a statement about one of the parts of the premise is true or false, which requires matching the number and the color-shape pair from the premise with those in the hypotheses or learning that 0 in the hypothesis with an attribute pair present in the premise makes the hypothesis incorrect. This is likely the reason why the model does not adapt to new attributes well, especially new attribute combinations, because it does not learn successfully those principles of processing information about attribute combinations, which does not depend on the combinations themselves.

4.3 Quantifiers 'no', 'some', 'all'

In the second stage of experiments we look at a different group of quantifiers, 'no', 'some' and 'all'. To correctly apply a quantifier to a scene in this setting, a different kind of reasoning from the case of numerical quantifiers has to be applied: it is necessary not only to judge how many objects have certain properties, but also how the number of objects that have certain properties relates to the total number of objects in a scene.

4.3.1 Main experiment

Values of hyperparameters selected during hyperparameter optimization: 30 LSTM units, 500 dense units, learning rate: 0.0023358024087867363.

The network hyperparameters were optimized using Bayesian optimization (with training set of size 64297, validation set of size 27557, test set of size 30618).

Evaluating on the test set:

Accuracy	Precision	Recall	F1 score
97.71	96.63	98.87	97.73

Table 4.9: Performance measures for selected hyperparameter values

Error analysis. Like in the first part of the experiments, precision is higher than recall, so most of the wrong predictions (75%) are due to false positives. 41% of examples that were incorrectly classified as positive have hypotheses with 'no' and all but 2 of those are about attribute pairs. Therefore the model does not recognize that the color-shape pair is present in the premise and because of the hypothesis with 'no' with this color-shape pair the example should be classified as negative. 23% of false positives have 'all' in the hypotheses, 87% of those examples are also about attribute pairs. The rest, 36% of false positives, have hypotheses with 'some', the overwhelming majority of them (95%) are attribute pair hypotheses.

There are only 173 false negatives. 14% have hypotheses with 'some', 22% have hypotheses with 'all', 64% have hypotheses with 'no'.

Overall the most problematic type of hypotheses were hypotheses with 'no' with attribute pairs where either the attribute pair was present in the premise and the hypothesis was therefore wrong, either the attribute pair was not in the premise but the shape was in the premise, and the hypothesis was true.

By looking at the results, we can answer **research subquestion 2.1**: the model can indeed learn to infer which statements with quantifiers 'no', 'some', 'all' are true and which are false given a complete description of a visual scene.

4.3.2 Exploring the influence of the amount of positive and negative examples

Answering **research subquestion 2.2**, accuracy peaks when the number of correct and incorrect hypotheses per premise in the training data data is 1. Precision increases with the increase of the number of incorrect hypotheses per premise and decreases with the increase of the number of correct hypotheses per premise. The reverse is true for recall: recall decreases with the increase of the number of incorrect hypotheses per premise and increases with the increase of the number of correct hypotheses per premise.

Num. of cor. and incor. hyp. per pr.	Accuracy	Precision	Recall	F1 score
3,1	82.16 ± 6.09	74.70 ± 6.83	98.81 ± 1.14	84.93 ± 4.58
2,1	81.31 ± 6.48	74.75 ± 7.20	96.20 ± 2.74	83.95 ± 4.92
1,1	85.37 ± 4.13	82.8 ± 4.08	89.46 ± 4.67	85.96 ± 3.90
1,2	85.02 ± 6.19	90.41 ± 3.72	78.08 ± 9.63	83.66 ± 7.03
1,3	84.59 ± 5.46	93.41 ± 3.12	74.31 ± 9.87	82.5 ± 6.73

Table 4.10: Results of experiment 2

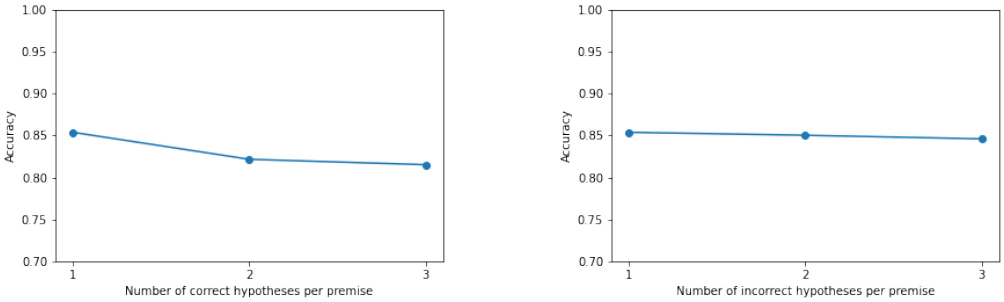


Figure 4.7: Accuracy depending on the number of correct and incorrect hypotheses per premise

4.3.3 Generalization experiments

Num.	Aspect	Accuracy	Precision	Recall	F1 score
1	Attributes	66.53 ± 2.02	68.25 ± 3.68	63.92 ± 13.69	64.83 ± 6.65
2a	Domain size	86.06 ± 5.2	84.23 ± 5.95	89.04 ± 4.74	86.51 ± 4.95
2b	Domain size	82.92 ± 3.69	81.63 ± 3.40	85.15 ± 6.11	83.24 ± 3.79
3	Attribute combinations	56.04 ± 1.85	57.63 ± 2.31	45.50 ± 2.77	50.83 ± 2.51
4	Form of premises	83.87 ± 3.65	82.82 ± 2.55	85.36 ± 5.52	84.04 ± 3.87

Table 4.11: Results of generalization experiments with quantifiers 'no', 'some', 'all'

1. **Attributes in the test set unseen during training.**

   As the training set 61236 randomly selected examples with old attributes were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'some objects are triangles', class: 0 (false). The model was tested on 61236 randomly selected examples with new attributes. An example of a premise-hypothesis pair from the test set: 'there are 3 green cones, there are 5 black cones, there are 6 green cones', 'all objects are cones', class: 1 (true).

   Error analysis.

Accuracy	Precision	Recall	F1 score
68.07	72.96	57.14	64.09

Table 4.12: Performance measures for a run of experiment 1

Precision is higher than recall. There are twice as many false negatives as there are false positives. 24% of false negatives are examples with hypotheses with 'no' and all of them are single attribute hypotheses. These hypotheses contain a shape or a color that is not present in the premise and capture the information that no objects in the scene have this color or shape. 50% of false negatives are examples with hypotheses with 'some', all of them contain attribute pairs. This means that with new unseen attributes the model does not recognize that an example where a premise is paired with a hypothesis with 'some' and a color-shape combination from one of the parts of this premise (indicating that this color-shape combination is present in the premise), then the example should be classified as correct. 26% of false negatives are examples with hypotheses with 'all', 95% of those are hypotheses with attribute pairs. In most of these examples, 90%, the shape appears in the premise once, so the hypothesis contains information only about one part of the premise, which means that in this case judging if a hypothesis with 'all' is true does not require combining information from different parts of the premise. In this situation judging if a hypothesis with 'all' is true or not is equivalent to evaluating a hypothesis with 'some'.

22% of false positives are examples with hypotheses with 'some', with single attributes. So examples with hypotheses stating that there are objects in the scene that have a certain color or shape while that is not true given the premise are incorrectly classified as true. 78% of false positives are examples with hypotheses with 'no' and all of them are with attribute pairs. These are hypotheses stating that a color-shape combination that is present in the premise is not present in the scene. None of the examples containing hypotheses with 'all' were incorrectly classified as positive.

The model struggles to process combinations of new unseen attributes and therefore does not learn to identify examples where attribute pairs present in the premise are in the hypotheses with quantifiers 'some' and 'all' as true; examples where single attributes not present in the premise are in the hypotheses with the quantifier 'no' as true; and examples where attribute combinations present in the premise are in the hypotheses with the quantifier 'no' as incorrect.

Based on the results, we can answer research **subquestion 2.3**. Similarly to the numerical quantifiers case, when predicting on data with quantifiers 'no', 'some', 'all' with attributes unseen during training, the model demonstrates sufficient performance but there is a noticeable decrease in accuracy.

2. **Domain size in the test set unseen during training.**

   a) As the training set 50000 randomly selected examples with domain size <= 15 were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'some objects are triangles', class: 0 (false). The model was tested on 50000 randomly selected examples with domain size > 15. An example of a premise-hypothesis pair from the test set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', 'all objects are circles', class: 0 (false).

   Error analysis.

Accuracy	Precision	Recall	F1 score
86.25	82.08	92.71	87.07

Table 4.13: Performance measures for a run of experiment 2a

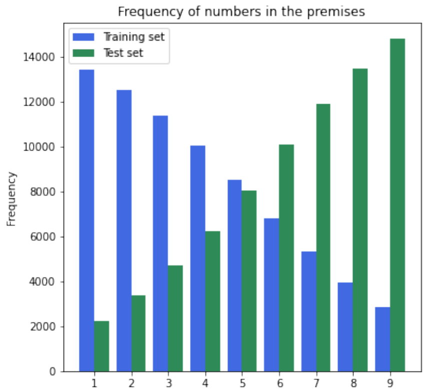


Figure 4.8: Frequency of numbers in premises in the training set and the test set for experiment 2a

In this experiment, contrary to the numerical quantifiers case, recall is higher than precision. There are twice more false positives than false negatives. 37% of false positives have hypotheses with 'no', 89% of those hypotheses are with attribute pairs. 98% of those attribute pair hypotheses contain color-shape combinations that are present in the premise, so in this case only reasoning about one of the parts of the premise is required, 67% of the attribute pairs have numbers higher than 5. 11% of the 'no' hypotheses are with single attributes, so again the model incorrectly considers 'no' hypotheses with attributes present in the premise as following from the premise. 36% of false positives have hypotheses with 'some' and 76% of those hypotheses are with attribute pairs, that are not present in the premise. 69% of those examples have premises where the shape is present and the numbers are higher than 5. The rest of false positives, 27%, have hypotheses with 'all', all with attribute pairs. In 67% of those examples the color and the shape in the hypothesis both appear in the premise, but not together, in 19% only the shape appears once in the premise. Again, the numbers associated with the shape or the color occurrence in the premise tend to be higher than 5, so this is a possible reason of wrong predictions.

As for false negatives, in 25% hypotheses are with 'all', mostly with attribute pairs 92%, most numbers next to the attribute pair are higher than 5. 41% hypotheses contain the quantifier 'some', all of them with attribute pairs. These examples again tend to have higher value numbers next to the attribute pairs in the premise. 34% hypotheses contain the quantifier 'no', 74% of those are with single attributes.

Overall the model struggles to identify if shapes and colors are present together in one of the parts of the premise when the numbers associated with those attributes are of higher value, as the model was not sufficiently exposed to this kind of numbers during training.

b) As the training set 50000 randomly selected examples with domain size > 15 were used. An example of a premise-hypothesis pair from the training set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', 'all objects are circles', class: 0 (false). The model was tested on 50000 randomly selected examples with domain size <= 15. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'some objects are triangles', class: 0 (false).

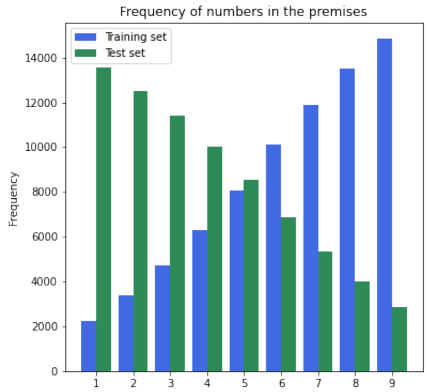


Figure 4.9: Frequency of numbers in premises in the training set and the test set for experiment 2b

Error analysis.

Accuracy	Precision	Recall	F1 score
80.88	79.26	83.47	81.31

Table 4.14: Performance measures for a run of experiment 2b

There are slightly more false positives than false negatives. 42% of false positives have hypotheses with 'no', all with attribute pairs. 47% of those attribute pairs are associated with a number lower than 5. 42% of false positives have hypotheses with 'some', 74% of which are with attribute pairs. These are hypotheses with attribute pairs not present in the premise, where the color and the shape appear in the premise, but separately, or only the shape appears in the premise, and attributes tend to be associated with lower value numbers that were not frequent in the training set. In 39% of those examples both the shape and the color appear in the premise, in 25% of those examples only the shape appears in the premise and in 14% of those examples the shape does not appear in the premise. The rest of false positives, 16%, are with 'all' hypotheses, all with attribute pairs that also are not in the premise. The shapes and colors out of theses attribute pairs appear in the premise separately mostly with numbers lower than 5.

36% of false negatives have hypotheses with 'all', 97% of which are with attribute pairs. In 46% of those examples the attribute pairs from the hypothesis are combined with a number lower than 5 in the premise. 33% of false negatives have hypotheses with 'no', 85% of them are with single attributes, that are not present in the premise and therefore the hypotheses are correct given the premises. 31% of false negatives have hypotheses with 'some', all of them with attribute pairs, again the numbers associated with the attribute pairs in the premise tend to be lower than 5.

When training on data in which numbers bigger than 5 are more frequent than numbers lower than 5 and testing on data for which the reverse is true, we see a similar pattern: most of the incorrect predictions are due to the model incorrectly handling attribute combinations when they are associated with numbers different from those that appeared frequently in the training set.

Based on the results of both parts of the experiment, we can answer **research subquestion 2.4**: the model generalizes well to data with quantifiers 'no', 'some', 'all' where domain size is unseen during training.

3. **Attribute combinations in the test set unseen during training.**

   As the training set 29160 examples with old color-shape combinations were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue triangles, there are 8 red circles, there are 3 yellow squares', 'no objects are triangles', class: 0 (false). The model was tested on 1458 examples with new color-shape combinations. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', 'some objects are red', class: 1 (true).

   Error analysis.

Accuracy	Precision	Recall	F1 score
58.23	61.03	45.54	52.16

Table 4.15: Performance measures for a run of experiment 3

   There are twice as many false negatives as there are false positives. Most false positives, 39%, are with the quantifier 'no', all but 3 with attribute pairs. Those are examples where the hypotheses state that an attribute combination is not present in the premise, while the premises do in fact contain those attribute combinations. 30% of false positives have hypotheses that are with the quantifier 'some', all with attribute pairs not present in the premise. In 87% of those examples both the color and the shape appear in the premise in different parts. 31% of false positives have hypotheses that are with the quantifier 'some', all also with attribute pairs not present in the premise.

   42% of false negatives are with the quantifier 'no', all with attribute pairs. Those attribute pairs do not appear in the premise, but in 88% of those examples the shape and the color appear in the premise separately, in different parts of the premise. 28% of false negatives are with the quantifier 'some', 99% with attribute pairs, so it is not recognized by the model that hypotheses stating that those attribute pairs are present in the premise are true. 30% of false negatives are with the quantifier 'all', again all with attribute pairs.

   We can see that, when predicting on data with attribute combinations unseen during training, the model fails to properly distinguish when colors and shapes appear together in the premise and when not and when a quantifier out of the set 'all', 'no' and 'some' is appropriate. It is noteworthy that we almost do not see false positive or false negative predictions for examples with hypotheses with single attributes, which means that with attribute combinations unseen during training the model has the ability to discern what shapes or colors are present or not present in the premise and what quantifier in the hypothesis applies to the situation.

   Answering **research subquestion 2.5**, the accuracy is still higher than 50% but it is quite low, so the model struggles to generalize to data with quantifiers 'no', 'some', 'all' where attribute combinations are unseen during training.

4. **Premises in the test set of the form unseen during training.**

   As the training set 61236 randomly selected examples with the old shape of premises were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', 'all objects are triangles', class: 0 (false). The model was tested on 61236 randomly selected examples with the new shape of premises. An example of a premise-hypothesis pair from the test set: 'there are 8 red squares, 8 red circles and 3 blue squares', 'some objects are red', class: 1 (true).

Error analysis.

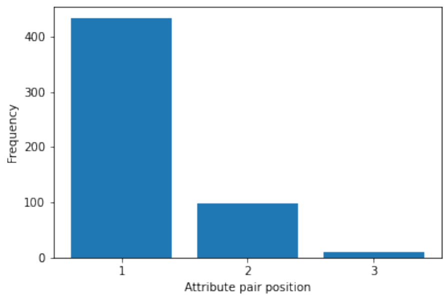
Accuracy	Precision	Recall	F1 score
86.21	84.87	88.27	86.53

Table 4.16: Performance measures for a run of experiment 4

Recall is higher than precision, so most of the wrong predictions are due to false positives. 44% of false positives have hypotheses with 'no', with attribute combinations that are present in the premise. 42% of false positives have hypotheses with 'some', 71% with attribute combinations that are not present in the premise. 14% of false positives have hypotheses with 'all', 98% with attribute pairs, either with the attribute combination not present in the premise, either with not all objects having the shape from the color-shape pair having this color, according to the premise.

45% of false negatives have hypotheses with 'no'. 57% of them have attribute combinations that are not present in the premise, 43% are with colors or shapes not present in the premise. 21% of false negatives have hypotheses with 'some', 85% with attribute combinations that are present in the premise. 34% of false negatives have hypotheses with 'all'.

We can see that there are two groups of examples that were classified incorrectly with attribute combinations from the hypothesis present in the premise: false negatives with hypotheses with 'some' and false positives with hypotheses with 'no'. To analyse if the position of attribute pairs in the premise influences the prediction, we build plots for these two groups.



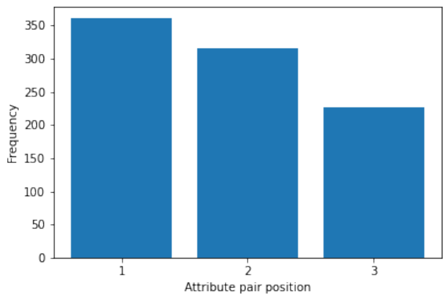


Figure 4.10: Number of examples with the attribute pair from the hypothesis present in a certain position in the premise for false negatives with hypotheses with 'some' and false positives with hypotheses with 'no'

It is evident that in both cases the highest number of the selected incorrectly classified examples contains the attribute pair that is in the hypothesis in the first position. This means that the model struggles to detect the existence of attribute pairs the most when they are placed in the first position, which is surprising given that between the training and test set it is the position of the second and the third attribute pair that changes.

Based on the results of the experiment, we can answer **research subquestion 2.6**: the model generalizes well to data with quantifiers 'no', 'some', 'all' where the form of premises is unseen during training.

4.3.4 Discussion

We can see that accuracy in the second part of experiments is very similar to accuracy in the first part of experiments involving numerical quantifiers when we look at all the experiments. The model again struggles with predicting on attributes unseen during training, with the lowest accuracy obtained in experiment 3, when color-shape combinations in the test data are different from the ones in the training data. Predicting on examples with domain size different from the domain size of training examples does not influence accuracy much, as well as predicting on data with the changed type of premises. We can see that the nature of prediction errors is slightly different between experiments 1 and 3: in experiment 3 there are almost no wrong predictions for examples that include single attribute hypotheses, so when the attributes themselves are completely changed between the training set and the test set, the model makes wrong predictions for both single attribute hypotheses and attribute pair hypotheses, but when the attribute set remains constant and only the attribute combinations change, the model makes wrong predictions only for attribute pair hypotheses. The comparison between experiment 1 and experiment 3 is not completely fair due to the limitations of the small test set in experiment 3, so we cannot make general conclusions based on these results.

4.4 Quantifiers 'more than n%', 'less than n%', 'exactly n%'

In the third stage of experiments we look at a wider range of quantifiers, quantifiers that reflect the proportion of objects having a certain property. The difference with the previous group of quantifiers is that in this part of experiments we explore proportional estimation more deeply with quantifiers 'exactly n%', with the addition of comparison reasoning in quantifiers 'more than n%', 'less than n%'. In the previous group of quantifiers the proportional estimation was only required to apply the quantifier 'all', that is now included in the form of 'exactly 100%'. Quantifiers 'some' and 'no' capturing the meaning of existence or non existence of objects having some attributes are also implicitly included in the new group of quantifiers: 'exactly 0%' = 'no', 'more than 0%' = 'some'. Important proportional quantifier 'most' can be seen as equivalent to 'more than 50%'.

4.4.1 Main experiment

Hyperparameter values selected during hyperparameter optimization: 100 LSTM units, 500 dense units, learning rate: 0.008401271006413085.

We then test the model with the selected hyperparameter values on the test set:

Accuracy	Precision	Recall	F1 score
86.76	86.48	86.94	86.71

Table 4.17: Performance measures for selected hyperparameter values

Error analysis.

Recall is slightly higher than precision, so more of the false predictions are false positives than false negatives. 33% of false positives have hypotheses with 'more than n%', 62% of those hypotheses are with attribute pairs. Hypotheses in these examples are either with an attribute pair that is present in the premise and the wrong percentage value, or with an attribute pair that is not present in the premise which makes the hypothesis incorrect (hypotheses with quantifiers 'more than n%' with all numbers $n = 0, ..., 99$ are incorrect for absent attributes). 35% of false positives have hypotheses with 'less than n%', 54% of those hypotheses are with attribute pairs. 65% of those hypotheses contain attribute pairs that are present in the premise and the percentage value is incorrect. 32% of false positives have hypotheses with 'exactly n%', 70% with single attributes. 72% of those hypotheses are with 'exactly 0%' with attributes present in the premise. 70% of hypotheses with attribute pairs are also with 'exactly 0%' with attributes present in the premise.

33% of false negatives have hypotheses with 'more than n%', 62% of those hypotheses are with attribute pairs. All of those attribute pairs are present in the premise with the correct percentage value. 35% of false negatives have hypotheses with 'less than n%', 54% of which are with attribute pairs. 70% of those hypotheses contain attribute pairs that are not present in the premise, which makes any 'less than' hypothesis correct. 60% of single attribute hypotheses with 'less than' contain attributes present in the premise with the correct percentage value. 32% of false negatives have hypotheses with 'exactly n%', 71% with single attributes. 73% of those hypotheses are with 0% and contain attributes not present in the premise.

Most false predictions occurred when the hypothesis contained a wrong percentage value or when it was not correctly identified if attributes were present in the premise.

Answering **research subquestion 3.1**, the model can indeed learn to infer which statements with quantifiers 'more than n%', 'less than n%', 'exactly n%' are true and which are false, but

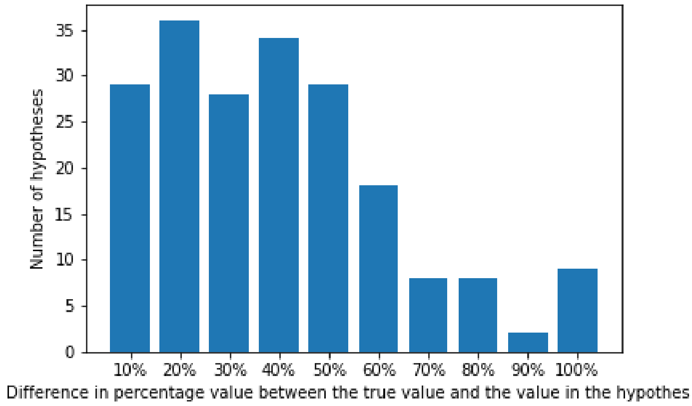


Figure 4.11: Number of examples with 'exactly n%' that were incorrectly classified as positive with different values of difference between the percentage value in the hypothesis and the true percentage value where $k - 10 < n <= k, n \neq 0$ for $k = 10, ..., 100$

with lower accuracy than for numerical quantifiers and quantifiers 'no', 'some', 'all'.

4.4.2 Exploring the influence of the amount of positive and negative examples

Num. of cor. and incor. hyp.	Accuracy	Precision	Recall	F1 score
3,1	76.15 ± 2.53	68.55 ± 2.60	96.96 ± 0.80	80.28 ± 1.66
2,1	79.25 ± 2.45	73.47 ± 3.18	92.13 ± 3.03	81.67 ± 1.80
1,1	81.07 ± 1.15	80.76 ± 2.29	81.86 ± 4.05	81.20 ± 1.44
1,2	79.87 ± 2.80	85.47 ± 2.70	72.18 ± 7.23	78.0 ± 4.23
1,3	74.40 ± 3.89	89.19 ± 2.98	55.93 ± 10.68	67.97 ± 7.46

Table 4.18: Results of experiment 2

Looking at the results, we can answer **research subquestion 3.2**. Like in the previous cases, accuracy peaks when the number of correct and incorrect hypotheses per premise is 1. Again precision is the highest when the number of incorrect hypotheses per premise is the highest and recall is the highest when the number of correct hypotheses per premise is the highest.

4.4.3 Generalization experiments

1. **Attributes in the test set unseen during training.**

   As the training set 61236 randomly selected examples with old attributes were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'exactly 35% of objects are triangles', class: 0 (false). The model was tested on 61236 randomly selected examples with new attributes. An example of a premise-hypothesis pair from the test set: 'there are 3 green cones, there

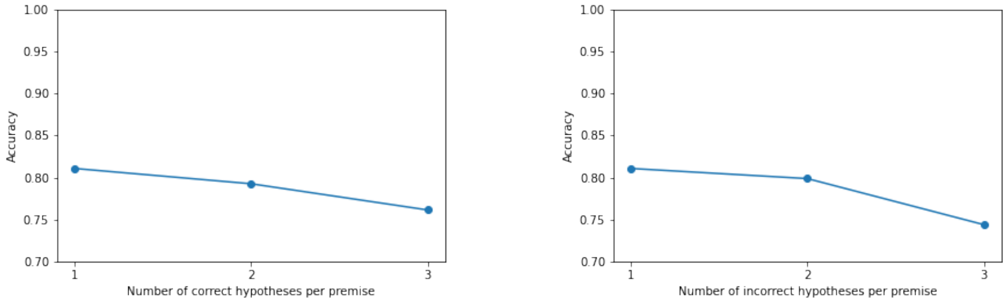


Figure 4.12: Accuracy depending on the number of correct and incorrect hypotheses per premise

Num.	Aspect	Accuracy	Precision	Recall	F1 score
1	Attributes	75.65 ± 1.02	75.73 ± 0.98	75.59 ± 4.16	75.58 ± 1.84
2a	Domain size	82.92 ± 2.38	81.52 ± 2.73	85.31 ± 3.90	83.31 ± 2.42
2b	Domain size	83.58 ± 2.34	83.19 ± 2.32	84.33 ± 5.49	83.63 ± 2.77
3	Attribute combinations	64.47 ± 2.37	67.36 ± 2.29	56.08 ± 5.11	61.10 ± 3.71
4	Form of premises	80.35 ± 2.40	79.02 ± 3.24	82.90 ± 4.76	80.78 ± 2.52

Table 4.19: Results of generalization experiments with quantifiers 'more than n%', 'less than n%', 'exactly n%'

are 5 black cones, there are 6 green cones', 'more than 90% of objects are cones', class: 1 (true).

Error analysis.

Accuracy	Precision	Recall	F1 score
76.53	75.57	78.35	76.94

Table 4.20: Performance measures for a run of experiment 2

Precision is lower than recall, there are two times more false positives than there are false negatives. 37% of false positives have hypotheses with 'less than n%', 78% of them are with attribute pairs. In 54% of them the attribute pair from the hypothesis is present in the premise and the percentage value in the hypothesis is not correct, in the rest the attribute pair is not present in the premise and the shape is not combined with a different color in the premise, so because of this the statement about this shape is not correct. In all hypotheses with single attributes the attribute is present in the premise and the percentage value in the hypothesis is not correct. 30% of false positives have hypotheses with 'more than n%', 63% of them are with attribute pairs. In 97% of them the attribute pair from the hypothesis is not present in the premise, which makes any 'more than' hypothesis incorrect. In 59% of the hypotheses with single attributes the attribute is not present in the premise. The rest, 33%, of hypotheses in false positives are with 'exactly n%', 87% of them with attribute pairs. In 91% of 'exactly' hypotheses with attribute pairs the color-shape combination from the hypothesis is present in the premise and the percentage value is incorrect. 89% of those hypotheses are with 'exactly 0%'. The number of examples incorrectly classified as positive with 'exactly n%' hypotheses, where $n \neq 0$, with different values of difference between the true value and the value in the hypothesis can be seen in Figure 4.13. There are more examples that have smaller difference between the correct value and the value in the hypothesis than examples that have bigger difference.

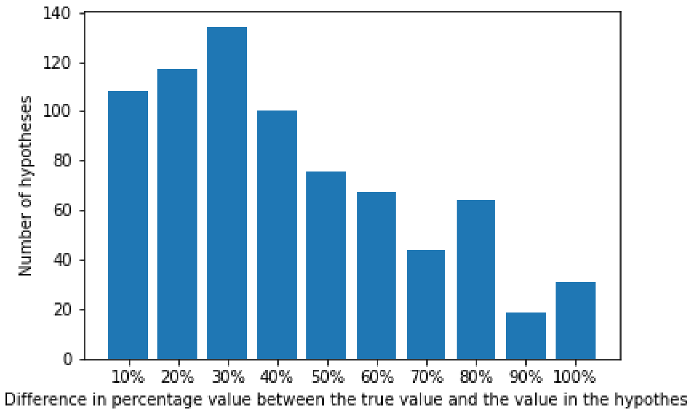


Figure 4.13: Number of examples with 'exactly n%' that were incorrectly classified as positive with different values of difference between the percentage value in the hypothesis and the true percentage value where $k - 10 < n <= k, n \neq 0$ for $k = 10, ..., 100$

31% of false negatives have hypotheses with 'less than n%', 53% of them are with attribute pairs. In 97% of them the attribute pair from the hypothesis is not present in the premise. 34% of false negatives have hypotheses with 'more than n%', 70% of them are with attribute pairs. In all of them the attribute pair from the hypothesis appears in the premise and the percentage value is correct. 35% of hypotheses in false negatives are with 'exactly n%', 89% of them with single attributes. 84% of those 'exactly' hypotheses are with '0%' with absent attributes.

Prediction errors here seem to be similar to prediction errors in the main experiment. The model especially struggles with 'exactly 0%' hypotheses.

Answering **research subquestion 3.3**, the model generalizes quite well to data with quantifiers 'more than n%', 'less than n%', 'exactly n%' where attributes are unseen during training.

2. **Domain size in the test set unseen during training.**

a) As the training set 50000 randomly selected examples with domain size <= 15 were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'more than 80% of objects are triangles', class: 0 (false). The model was tested on 50000 randomly selected examples with domain size > 15. An example of a premise-hypothesis pair from the test set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', 'exactly 100% of objects are circles', class: 0 (false).

Error analysis.

Precision is lower than recall, so there are more false positives than false negatives. 35% of false positives have hypotheses with 'less than n%', 67% of them are with attribute pairs. In 56% of them attribute pairs are present in the premise and the percentage value

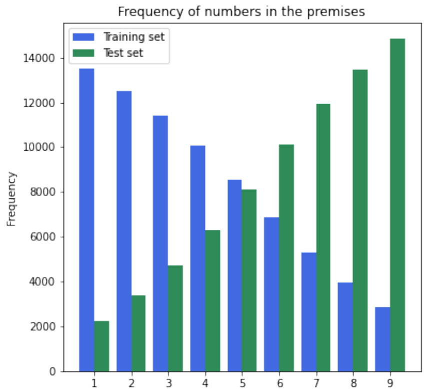


Figure 4.14: Frequency of numbers in premises in the training set and the test set for experiment 3a

Accuracy	Precision	Recall	F1 score
82.73	81.42	84.81	83.08

Table 4.21: Performance measures for a run of experiment 3a

is not correct, for 70% the attribute pair is associated with a number higher than 5. In the remaining 44% of examples the attribute pair does not appear in the premise and the shape is not in the premise, so any 'less than' hypothesis is incorrect. 34% of false positives have hypotheses with 'more than n%', 61% of them are with attribute pairs. In 74% of those hypotheses with attribute pairs the attribute pairs are not present in the premise, which makes any 'more than' hypothesis wrong. 31% of false positives have hypotheses with 'exactly n%', 61% of them are with attribute pairs, 88% of them present in the premise. Most of those attributes are combined with a high value number in the premise.

41% of false negatives have hypotheses with 'less than n%', 66% of them are with attribute pairs. In 72% of them attribute pairs are not present in the premise, so the hypothesis with 'less' is correct not depending on the numbers in the premise. In 67% of the examples with hypotheses with attribute pairs that are present in the premise, the attribute pair was associated with a number higher than 5. 27% of false negatives have hypotheses with 'more than n%', 71% of them are with attribute pairs. All of those attribute pairs are present in the premise, so the percentage value is correct. In 69% of those examples the attribute pair was associated with a number higher than 5. 32% of false negatives have hypotheses with 'exactly n%', 54% of them are with single attributes, 68% of them are with 'exactly 0%' with attributes not present in the premise. 85% of 'exactly' hypotheses with attribute pairs in the examples incorrectly classified as negative are with 'exactly 0%' with attribute pairs not present in the premise.

In general, we can see that mistakes in predictions are mostly due to the model not being able to handle numbers that were less frequent in the training data than in the test data (numbers higher than 5) and incorrectly identify when an attribute pair is present or not in the premise.

b) As the training set 50000 randomly selected examples with domain size > 15 were used. An example of a premise-hypothesis pair from the training set: 'there are 8 blue circles, there are 8 yellow circles, there are 8 red squares', 'exactly 100% of objects are circles', class: 0 (false). The model was tested on 50000 randomly selected examples with domain size <= 15. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 yellow circles, there are 3 red squares', 'more than 80% of objects are triangles', class: 0 (false).

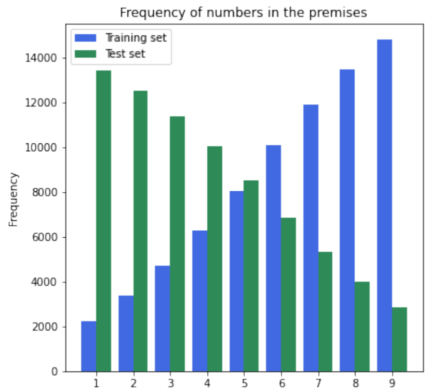


Figure 4.15: Frequency of numbers in premises in the training set and the test set for experiment 3b

Error analysis.

Accuracy	Precision	Recall	F1 score
82.61	80.76	85.63	83.12

Table 4.22: Performance measures for a run of experiment 3b

Precision is lower than recall, so there are more false positives than false negatives. 35% of false positives have hypotheses with 'less than n%', 60% of them are with attribute pairs. In 96% of them attribute pairs are present in the premise and the percentage value is not correct. For 53% the attribute pair is associated with a number lower than 5. 25% of false positives have hypotheses with 'more than n%', 66% of them are with attribute pairs. In 78% of those hypotheses the attribute pair is not present in the premise which makes any 'more' hypothesis incorrect. 40% of false positives have hypotheses with 'exactly n%', 67% of them are with attribute pairs, 96% of them present in the premise and the hypothesis is with 'exactly 0%'. Most of those attributes are combined with a lower value number in the premise.

42% of false negatives have hypotheses with 'more than n%', 55% of them are with attribute pairs. In all of them attribute pairs are present in the premise, so the percentage value in the hypothesis is correct. In 48% of the examples with hypotheses with attribute pairs that are present in the premise, the attribute pair was associated with a number lower than 5. 88% of the single attribute hypotheses in false negatives with 'more than n%' contain attributes present in the premise and the percentage values is also not correct. 31% of false negatives have hypotheses with 'less than $n\%$', 68% of them are with attribute pairs. 94% of those attribute pairs are not present in the premise, so a hypothesis with 'less' is always correct in this case. 27% of false negatives have hypotheses with 'exactly $n\%$', 58% of them are with single attributes, 55% of them are present in the premise. The remaining 45% of those single attribute hypotheses are with 'exactly 0%'.

The influence of numbers in the premises that are less frequent in the test data than in the training data in this experiment is less evident than in the previous one. Most of the prediction mistakes can be attributed to failure in determining presence or absence of attributes from the hypothesis in the premise.

Based on the results of both parts of the experiment, we can answer **research subquestion 3.4**: the model generalizes well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where domain size is unseen during training.

3. **Attribute combinations in the test set unseen during training.**

   As the training set 29160 examples with old color-shape combinations were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue triangles, there are 8 red circles, there are 3 yellow squares', 'more than 50% of objects are triangles', class: 0 (false). The model was tested on 1458 examples with new color-shape combinations. An example of a premise-hypothesis pair from the test set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', 'less than 60% of objects are red', class: 1 (true).

   Error analysis.

Accuracy	Precision	Recall	F1 score
63.17	64.41	58.85	61.51

Table 4.23: Performance measures for a run of experiment 4

Precision is higher than recall, so there are more false negatives than false positives. 37% of false negatives have hypotheses with 'more than $n\%$', 78% of them are with attribute pairs. In all of them attribute pairs are present in the premise, so the percentage value in the hypothesis is correct. 31% of false negatives have hypotheses with 'less than $n\%$', 94% of them are with attribute pairs. All of those attribute pairs are not present in the premise, so a hypothesis with 'less' is always correct in this case. 32% of false negatives have hypotheses with 'exactly $n\%$', 95% of them are with attribute pairs, 84% of them are not present in the premise and the hypothesis is with 'exactly 0%'.

35% of false positives have hypotheses with 'less than $n\%$', 56% of them are with attribute pairs. All of those attribute pairs are present in the premise and any 'less' hypothesis is incorrect, because, since there are only 3 color-shape combinations in the test data, the shape can only appear in the premise together with the color from the hypothesis and the correct percentage value is always 100%. For 92% of single attribute hypotheses in false positives the attribute is present in the premise and the percentage value is incorrect. 28% of false positives have hypotheses with 'more than $n\%$', 57% of them are with attribute pairs. In all of those hypotheses the attribute pair is not present in the premise, which makes any 'more' hypothesis incorrect. 37% of false positives have hypotheses with 'exactly $n\%$', 65% of them are with attribute pairs, in 93% of them attribute pairs are present in the premise, so the percentage value is incorrect.

Most prediction errors seem to come from the model being unable to detect attribute combinations that are unseen during training.

Answering **research subquestion 3.5**, the model's performance is sufficient but accuracy is quite low when testing on data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where attribute combinations are unseen during training.

4. **Premises in the test set of the form unseen during training.**

   As the training set 61236 randomly selected examples with the old shape of premises were used. An example of a premise-hypothesis pair from the training set: 'there are 3 blue circles, there are 8 red squares, there are 3 yellow triangles', 'less than 60% of triangles are yellow', class: 0 (false). The model was tested on 61236 randomly selected examples with the new shape of premises. An example of a premise-hypothesis pair from the test set: 'there are 8 red squares, 8 red circles and 3 blue squares', 'more than 25% of objects are red', class: 1 (true).

   Error analysis

Accuracy	Precision	Recall	F1 score
80.53	77.48	86.0	81.52

Table 4.24: Performance measures for a run of experiment 5

Precision is lower than recall, so there are more false positives than false negatives. 31% of false positives have hypotheses with 'less than $n\%$', 59% of them are with attribute pairs. 99.9% of those attribute pairs are present in the premise and the percentage value is not correct. 95% of the hypotheses with single attributes contain attributes that are present in the premise and the percentage value is also not correct. 29% of false positives have hypotheses with 'more than $n\%$', 57% of them are with attribute pairs. In 85% of those hypotheses the attribute pair is not present in the premise which makes any 'more' hypothesis incorrect. 40% of false positives have hypotheses with 'exactly $n\%$', 72% of them are with attribute pairs, 95% of them present in the premise and the percentage value is incorrect. 89% of those hypotheses are with 'exactly 0%'. The number of examples incorrectly classified as positive with 'exactly $n\%$' hypotheses, where $n \neq 0$, with different values of difference between the true value and the value in the hypothesis can be seen in Figure 4.16.

40% of false negatives have hypotheses with 'more than $n\%$', 67% of them are with attribute pairs. In all of them attribute pairs are present in the premise, so the percentage value in the hypothesis is correct. 32% of false negatives have hypotheses with 'less than $n\%$', 59% of them are with attribute pairs. 94% of those attribute pairs are not present in the premise, so a hypothesis with 'less' is always correct in this case. 28% of false negatives have hypotheses with 'exactly $n\%$', 80% of them are with single attributes, 74% of them are present in the premise.

Looking at the results of the experiment, we can answer **research subquestion 3.6**: the model generalizes well to data with quantifiers 'more than $n\%$', 'less than $n\%$', 'exactly $n\%$' where the form of premises is unseen during training.

4.4.4 Discussion

Accuracy in the main experiment is lower than in part 1 and 2 of the experiments. This is not unexpected, as the group of quantifiers is now more challenging and requires proportional

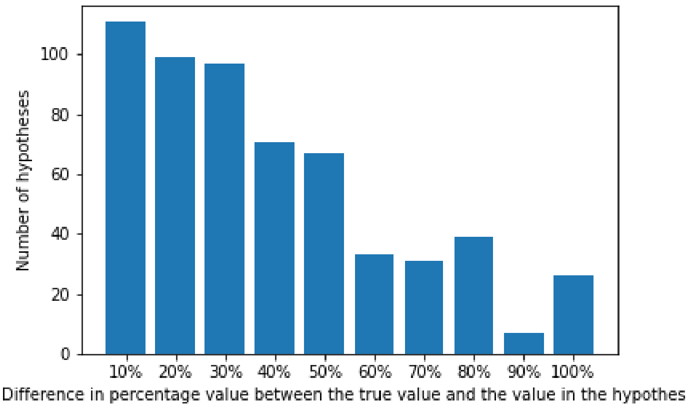


Figure 4.16: Number of examples with 'exactly n%' that were incorrectly classified as positive with different values of difference between the percentage value in the hypothesis and the true percentage value where $k - 10 < n <= k, n \neq 0$ for $k = 10, ..., 100$

estimation and comparison. However, accuracy in generalization experiments 1 and 3 is higher than accuracy in the same experiments in part 1 and 2. This might mean that the model was able to adapt to unseen attributes better.

Accuracy in all other experiments is similar to accuracy in the same experiments in previous parts. It is noteworthy that the model struggles specifically with hypotheses containing the quantifier 'exactly 0%', so it does not learn to associate it with absent attributes. Prediction errors show that one source of errors is the fact that attributes from the hypothesis are present or missing in the premise is often not correctly identified by the model and another source of errors is incorrect handling of percentage values: classifying an example with a hypothesis that contains a wrong percentage value as positive or classifying an example with a hypothesis that contains a correct percentage value as negative.

Chapter 5

Conclusion

In all three parts of our experiments we see the same pattern: the model performs well in the first experiment on the initial dataset and the performance drops in experiments evaluating generalization ability of the model. The performance on the initial dataset is quite remarkable given the simple architecture of the model. Accuracy in all generalization experiments with the exception of the experiment with testing on data with unseen color-shape combinations remains quite high and serves as an indication that the model learns some underlying principles of associating different types of quantifiers with descriptions of visual scenes. Testing on data with attributes or attribute combinations that did not appear in the training data notably leads to lower accuracy than testing on data with descriptions of scenes having a different domain size or data with a changed form of premises. This shows that the quality of prediction of the model depends more on being exposed to attributes or attribute combinations similar to the ones in the test data during training as opposed to being exposed to combinations of numbers in premises similar to the ones in the test data during training or the structure of premises being similar.

The first two groups of quantifiers that we explored, numerical quantifiers and quantifiers 'all', 'some', 'no' were of similar semantic complexity. These groups were of similar semantic complexity and did not require proportional estimation. The first group required just counting but the second one already required some analysis to establish if all, some or no objects from a set have a certain property. The last group of quantifiers was more advanced and required proportional estimation along with comparison.

It is noteworthy that the performance of the model is quite similar for all three groups of quantifiers that we explored in this project, if we consider the variety of their semantic and logical properties. It is easier to tell if the quantifier 'some' applies to a situation than to tell if the quantifier 'more than 35%' holds. Results of part 1 and part 2 are very close with the difference in accuracy for all experiments being around 2-3 percentage points, with the exception of generalization experiment 3, where the difference is 5 percentage points. In the third part of experiments we see a little different picture: accuracy on the initial experiment is lower than in part 1 and 2, as is accuracy in the last experiment, with testing on data with an unseen type of premises. The performance in the two part experiment with testing on data with a domain size that is different from the domain size values in the training data is similar to part 1 and 2. However, the performance in the two most challenging experiments, generalization experiment 1 with testing on data with a new set of attributes unseen during training and experiment 3 with testing on data with a new set of attribute combinations unseen during training, is the best out of all three groups of quantifiers. A possible explanation for this is the complexity of the last group of quantifiers. It is the most varied group that encapsulates the meaning of a bigger array of quantifiers, so possibly training the model on data containing more diverse linguistic constructs makes it more likely to make correct predictions.

Based on the results of our experiments, we can conclude that, given a complete description of a visual scene, simple neural networks are in fact able to learn to tell whether a quantified statement about the objects in the scene is true or false. This also answers another question that we posed about the applicability of our approach to Visual Inference. Our model can be used a second component of a Visual Inference system with the first component producing textual descriptions of the visual data. This approach might yield interesting new results for the Visual Inference task.

We have to note, however, that our research has some shortcomings. These points can serve as inspiration for further research in the area.

1. First and foremost, to investigate if simple neural networks can exhibit true understanding of logical aspects of quantification instead of relying on shallow heuristics, interpretability research is a necessary next step.

2. We did not investigate the influence of the distribution of the hypotheses in the generated datasets among different types on the performance of the model. We did not try to balance out the amount of hypotheses with a certain quantifier that follow from the premise with the amount of hypotheses with the same quantifier that do not follow from the premise. A possible direction of further research would be to explore this deeper.

3. We did not experiment with different word embeddings. It would be interesting to try embeddings that capture numeracy in further research.

4. In general it would be interesting to explore quantification on more varied data and also explore more types of quantifiers.

Bibliography

[1] Lasha Abzianidze. “A tableau prover for natural logic and language”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2492–2502.

[2] Lasha Abzianidze. “Natural solution to FraCaS entailment problems”. In: *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. 2016, pp. 64–74.

[3] Manoj Acharya, Kushal Kafle, and Christopher Kanan. “TallyQA: Answering complex counting questions”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8076–8084.

[4] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. “Analyzing the behavior of visual question answering models”. In: *arXiv preprint arXiv:1606.07356* (2016).

[5] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. “Giving bert a calculator: Finding operations and arguments with reading comprehension”. In: *arXiv preprint arXiv:1909.00109* (2019).

[6] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. “Neural module networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 39–48.

[7] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. “Vqa: Visual question answering”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[8] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. “Counting in the wild”. In: *European conference on computer vision*. Springer. 2016, pp. 483–498.

[9] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. “See, hear, and read: Deep aligned representations”. In: *arXiv preprint arXiv:1706.00932* (2017).

[10] Jorge A Balazs, Edison Marrese-Taylor, Pablo Loyola, and Yutaka Matsuo. “Refining raw sentence representations for textual entailment recognition via attention”. In: *arXiv preprint arXiv:1707.03103* (2017).

[11] Jon Barwise and Robin Cooper. “Generalized quantifiers and natural language”. In: *Philosophy, language, and artificial intelligence*. Springer, 1981, pp. 241–301.

[12] Luisa Bentivogli, Elena Cabrio, Ido Dagan, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. “Building Textual Entailment Specialized Data Sets: a Methodology for Isolating Linguistic Phenomena Relevant to Inference.” In: *LREC*. 2010.

[13] Samuel Bowman, Christopher Potts, and Christopher D Manning. “Recursive neural networks can learn logical semantics”. In: *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*. 2015, pp. 12–21.

[14] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. “A large annotated corpus for learning natural language inference”. In: *arXiv preprint arXiv:1508.05326* (2015).

[15] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. “Counting everyday objects in everyday scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1135–1144.

[16] Guanyi Chen, Kees van Deemter, and Chenghua Lin. “Generating Quantified Descriptions of Abstract Visual Scenes”. In: *Proceedings of the 12th International Conference on Natural Language Generation*. Tokyo, Japan: Association for Computational Linguistics, Oct. 2019, pp. 529–539. DOI: 10.18653/v1/W19-8667. URL: https://www.aclweb.org/anthology/W19-8667.

[17] Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. “Question Directed Graph Attention Network for Numerical Reasoning over Text”. In: *arXiv preprint arXiv:2009.07448* (2020).

[18] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. “Enhanced lstm for natural language inference”. In: *arXiv preprint arXiv:1609.06038* (2016).

[19] Peter Clark. “What Knowledge is Needed to Solve the RTE5 Textual Entailment Challenge?” In: *arXiv preprint arXiv:1806.03561* (2018).

[20] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. “Supervised learning of universal sentence representations from natural language inference data”. In: *arXiv preprint arXiv:1705.02364* (2017).

[21] Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. *Using the framework*. Tech. rep. Technical Report LRE 62-051 D-16, The FraCaS Consortium, 1996.

[22] Ido Dagan, Oren Glickman, and Bernardo Magnini. “The pascal recognising textual entailment challenge”. In: *Machine Learning Challenges Workshop*. Springer. 2005, pp. 177–190.

[23] Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. “Finding contradictions in text”. In: *Proceedings of ACL-08: HLT*. 2008, pp. 1039–1047.

[24] Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. “Guesswhat?! visual object discovery through multi-modal dialogue”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5503–5512.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[26] Yubing Dong, Ran Tian, and Yusuke Miyao. “Encoding generalized quantifiers in dependency-based compositional semantics”. In: *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*. 2014, pp. 585–594.

[27] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. “DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs”. In: *arXiv preprint arXiv:1903.00161* (2019).

[28] Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. “Stress-testing neural models of natural language inference with multiply-quantified sentences”. In: *arXiv preprint arXiv:1810.13033* (2018).

[29] Bart Geurts, Napoleon Katsos, Chris Cummins, Jonas Moons, and Leo Noordman. "Scalar quantifiers: Logic, acquisition, and processing". In: *Language and cognitive processes* 25.1 (2010), pp. 130–148.

[30] Mor Geva, Ankit Gupta, and Jonathan Berant. "Injecting numerical reasoning skills into language models". In: *arXiv preprint arXiv:2004.04487* (2020).

[31] Patrice Hartnett and Rochel Gelman. "Early understandings of numbers: Paths or barriers to the construction of new understandings?" In: *Learning and instruction* 8.4 (1998), pp. 341–374.

[32] Izumi Haruta, Koji Mineshima, and Daisuke Bekki. *Logical Inferences with Comparatives and Generalized Quantifiers.* 2020. arXiv: `2005.07954 [cs.CL]`.

[33] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[34] Micah Hodosh, Peter Young, and Julia Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics". In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 853–899.

[35] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. "Learning to solve arithmetic word problems with verb categorization". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 2014, pp. 523–533.

[36] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. "Convolutional neural network architectures for matching natural language sentences". In: *Advances in neural information processing systems.* 2014, pp. 2042–2050.

[37] Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S Moss, and Sandra Kübler. "Monalog: a lightweight system for natural language inference based on monotonicity". In: *arXiv preprint arXiv:1910.08772* (2019).

[38] Danqing Huang, Shuming Shi, Chin-Yew Lin, and Jian Yin. "Learning fine-grained expressions to solve math word problems". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.* 2017, pp. 805–814.

[39] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. "Visual storytelling". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2016, pp. 1233–1239.

[40] Felicia Hurewitz, Anna Papafragou, Lila Gleitman, and Rochel Gelman. "Asymmetries in the Acquisition of Numbers and Quantifiers". In: *Language Learning and Development* 2 (Apr. 2006), pp. 77–96. DOI: `10.1207/s15473341lld0202_1`.

[41] Thomas F Icard III and Lawrence S Moss. "Recent progress on monotonicity". In: *Linguistic Issues in Language Technology, Volume 9, 2014-Perspectives on Semantic Representations for Textual Inference.* 2014.

[42] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017, pp. 2901–2910.

[43] Kushal Kafle and Christopher Kanan. “An analysis of visual question answering algorithms”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1965–1973.

[44] Kushal Kafle and Christopher Kanan. “Answer-type prediction for visual question answering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4976–4984.

[45] Kushal Kafle and Christopher Kanan. “Visual question answering: Datasets, algorithms, and future challenges”. In: *Computer Vision and Image Understanding* 163 (2017), pp. 3–20.

[46] William A Ladusaw. “Polarity Sensitivity as Inherent Scope Relations.” In: (1980).

[47] Alice Yingming Lai. “Textual entailment from image caption denotations”. PhD thesis. University of Illinois at Urbana-Champaign, 2018.

[48] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. “Program induction by rationale generation: Learning to solve and explain algebraic word problems”. In: *arXiv preprint arXiv:1705.04146* (2017).

[49] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. “A SICK cure for the evaluation of compositional distributional semantic models.” In: *Lrec*. Reykjavik. 2014, pp. 216–223.

[50] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. “Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference”. In: *arXiv preprint arXiv:1902.01007* (2019).

[51] Koji Mineshima, Pascual Martınez-Gómez, Yusuke Miyao, and Daisuke Bekki. “Higher-order logical inference with compositional semantics”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2055–2061.

[52] Arindam Mitra and Chitta Baral. “Learning to use formulas to solve simple arithmetic problems”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 2144–2153.

[53] Richard Montague. “The proper treatment of quantification in ordinary English”. In: *Approaches to natural language*. Springer, 1973, pp. 221–242.

[54] Andrzej Mostowski. “On a generalization of quantifiers”. In: *Fundamenta mathematicae* 44.2 (1957).

[55] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. “Natural language inference by tree-based convolution and heuristic matching”. In: *arXiv preprint arXiv:1512.08422* (2015).

[56] Jonas Mueller and Aditya Thyagarajan. “Siamese recurrent architectures for learning sentence similarity”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[57] Mathijs S Mul. “Recognizing Logical Entailment: Reasoning with Recursive and Recurrent Neural Networks”. In: (2018).

[58] Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. “Stress test evaluation for natural language inference”. In: *arXiv preprint arXiv:1806.00692* (2018).

[59] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. “Revisiting Modulated Convolutions for Visual Counting and Beyond”. In: *arXiv preprint arXiv:2004.11883* (2020).

[60] Yixin Nie and Mohit Bansal. "Shortcut-stacked sentence encoders for multi-domain inference". In: *arXiv preprint arXiv:1708.02312* (2017).

[61] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. "A decomposable attention model for natural language inference". In: *arXiv preprint arXiv:1606.01933* (2016).

[62] Sandro Pezzelle, Marco Marelli, and Raffaella Bernardi. "Be precise or fuzzy: Learning the meaning of cardinals and quantifiers from vision". In: *arXiv preprint arXiv:1702.05270* (2017).

[63] Sandro Pezzelle, Ionut-Teodor Sorodoc, and Raffaella Bernardi. "Comparatives, quantifiers, proportions: a multi-task model for the learning of quantities from vision". In: *arXiv preprint arXiv:1804.05018* (2018).

[64] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving language understanding by generative pre-training". In: (2018).

[65] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. "NumNet: Machine reading comprehension with numerical reasoning". In: *arXiv preprint arXiv:1910.06701* (2019).

[66] Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. "EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference". In: *arXiv preprint arXiv:1901.03735* (2019).

[67] Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. "Probing natural language inference models through semantic fragments". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8713–8721.

[68] Subhro Roy, Tim Vieira, and Dan Roth. "Reasoning about quantities in natural language". In: *Transactions of the Association for Computational Linguistics* 3 (2015), pp. 1–13.

[69] Mark Sammons, VG Vinod Vydiswaran, and Dan Roth. "Ask not what Textual Entailment can do for You..." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010, pp. 1199–1208.

[70] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. "A simple neural network module for relational reasoning". In: *arXiv preprint arXiv:1706.01427* (2017).

[71] Santi Seguı, Oriol Pujol, and Jordi Vitria. "Learning to count with deep object features". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2015, pp. 90–96.

[72] Ionut Sorodoc, Angeliki Lazaridou, Gemma Boleda, Aurélie Herbelot, Sandro Pezzelle, and Raffaella Bernardi. ""Look, some green circles!": Learning to quantify from images". In: *Proceedings of the 5th Workshop on Vision and Language*. 2016, pp. 75–79.

[73] Ivilin Stoianov and Marco Zorzi. "Emergence of a'visual number sense'in hierarchical generative models". In: *Nature neuroscience* 15.2 (2012), pp. 194–196.

[74] Jakub Szymanik and Camilo Thorne. "Exploring the relation between semantic complexity and quantifier distribution in large corpora". In: *Language Sciences* 60 (2017), pp. 80–93.

[75] Alfred Tarski. "The concept of truth in the languages of the deductive sciences". In: *Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznych* 34.13-172 (1933), p. 198.

[76] Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. "Logical inference on dependency-based compositional semantics". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 79–89.

[77] Alexander Trott, Caiming Xiong, and Richard Socher. "Interpretable counting for visual question answering". In: *arXiv preprint arXiv:1712.08697* (2017).

[78] Shyam Upadhyay, Ming-Wei Chang, Kai-Wei Chang, and Wen-tau Yih. "Learning from explicit and implicit supervision jointly for algebra word problems". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 297–306.

[79] Johan Van Benthem et al. *Essays in logical semantics*. Springer, 1986.

[80] Ton Van der Wouden. *Negative contexts: Collocation, polarity and multiple negation*. Routledge, 2002.

[81] Savvas Varsamopoulos, Koen Bertels, and Carmen Garcia Almudever. "Comparing neural network based decoders for the surface code". In: *IEEE Transactions on Computers* 69.2 (2019), pp. 300–311.

[82] Sara Veldhoen and Willem Zuidema. "Can neural networks learn logical reasoning?" In: *CLASP Papers in Computational Linguistics* (2017), p. 34.

[83] Hoa Trong Vu, Claudio Greco, Aliia Erofeeva, Somayeh Jafaritazehjan, Guido Linders, Marc Tanti, Alberto Testoni, Raffaella Bernardi, and Albert Gatt. "Grounded textual entailment". In: *arXiv preprint arXiv:1806.05645* (2018).

[84] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. "Do nlp models know numbers? probing numeracy in embeddings". In: *arXiv preprint arXiv:1909.07940* (2019).

[85] Jianyu Wang, Zhishuai Zhang, Cihang Xie, Yuyin Zhou, Vittal Premachandran, Jun Zhu, Lingxi Xie, and Alan Yuille. "Visual concepts and compositional voting". In: *arXiv preprint arXiv:1711.04451* (2017).

[86] Shuohang Wang and Jing Jiang. "Learning natural language inference with LSTM". In: *arXiv preprint arXiv:1512.08849* (2015).

[87] Adina Williams, Nikita Nangia, and Samuel R Bowman. "A broad-coverage challenge corpus for sentence understanding through inference". In: *arXiv preprint arXiv:1704.05426* (2017).

[88] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. "Visual entailment: A novel task for fine-grained image understanding". In: *arXiv preprint arXiv:1901.06706* (2019).

[89] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. "Do Neural Models Learn Systematicity of Monotonicity Inference in Natural Language?" In: *arXiv preprint arXiv:2004.14839* (2020).

[90] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. "Can neural networks understand monotonicity reasoning?" In: *arXiv preprint arXiv:1906.06448* (2019).

[91] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. "HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning". In: *arXiv preprint arXiv:1904.12166* (2019).

[92] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. "Cross-scene crowd counting via deep convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 833–841.

[93] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. "Salient object subitizing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4045–4054.

[94] Senlin Zhang, Siyang Liu, and Meiqin Liu. "Natural language inference using LSTM model with sentence fusion". In: *2017 36th Chinese Control Conference (CCC)*. IEEE. 2017, pp. 11081–11085.

[95] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. "Learning to count objects in natural images for visual question answering". In: *arXiv preprint arXiv:1802.05766* (2018).

[96] Lipu Zhou, Shuaixiang Dai, and Liwei Chen. "Learn to solve algebra word problems using quadratic programming". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 817–822.