Universiteit Leiden

Master Computer Science

Dataset Creation
for Visual Entailment
using Generative AI

Name:	Rob Reijtenbach
Student ID:	s1568159
Date:	11/07/2024
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Gijs Wijnholds
2nd supervisor:	Prof. dr. Suzan Verberne

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Within the field of natural language processing, *natural language inference*, sometimes called textual entailment, is a classification problem in which a premise-hypothesis pair has to be given a label, usually entailment, neutral or contradiction. More recently, built on the idea of textual entailment, there is visual-textual entailment or visual entailment for short. In visual entailment the premise is substituted by an image making the premise-hypothesis pair consist of an premise image and a hypothesis text. In order to train models to correctly classify image-hypothesis pairs, datasets are needed.

While datasets of images combined with hypotheses and labels already exist, e.g. SNLI-VE, there are several other datasets for conventional, textual entailment. In this work we research the viability of using generative AI to generate images for the premises of a textual entailment dataset in order to create a visual entailment dataset. This is done by generating synthetic versions of the SNLI-VE and SICK-VTE dataset and conducting experiments on these generated datasets. The goal of this is to be able to create more visual entailment datasets out of the existing textual datasets.

We broadly execute three different experiments. In the first experiment we look at the intrinsic similarity of generated images compared to original images, the second investigates how well a model trained on generated images performs compared to a model trained on original data. The third category of experiments focuses on transfer learning of trained models to an entirely new dataset. This entails taking the models that were trained on one dataset and its generated version and comparing their performance when evaluating on a different dataset.

The results of the first experiment suggest that the generated images are similar to the original images they are based on as when ordering the generated images by similarity to the original, the images based on that original image usually end up high in the ranked list. The results of the second experiment show an accuracy of 68.8% for a model trained on generated images compared to 70.3% for a model that is trained on real images, both evaluated on real images, which suggests only slightly lower data utility for the generated data compared to the original data. The third experiment shows an accuracy of $\sim$50% when both of the trained models are evaluated on a new dataset, in this case SICK-VTE, which is similar accuracy to random guessing as this dataset only has two output classes. This is not really indicative of viability of using generated data as the performance of the model trained on original data performs only marginally better than the one trained on generated images. In conclusion, we are positive about the viability of using generative models for visual entailment dataset creation.

Contents

Acknowledgement

I would like to extend my gratitude to Dr. Gijs Wijnholds for his guidance and supervision throughout this project and for the feedback and inspiration that have greatly helped me throughout the past eight months. Him introducing me to the fields of natural language inference and visual entailment is what started this project and his help allowed me to bring it to a successful conclusion. I would also like to thank Prof. dr. Suzan Verberne, not only for her help as second supervisor in finalizing my thesis but also for being available for questions throughout my project as well.

Furthermore, I want to thank my family for supporting me throughout, not only this project, but my academic journey as a whole and my beloved girlfriend for helping me stay motivated to finish this project on time. Finally, I want to thank my friends, who helped me relax when I was not working on my thesis and in particular, Elgar van der Zande, without whose firm but constructive criticism over the years, my programming skills would not have been at the level they are at today.

Chapter 1

Introduction

In the field of computational *natural language processing*, one of the tasks is *natural language inference*. Bos and Markert [2005] call natural language inference is *"one of the ultimate challenges for any NLP system"*.[1] They state that a system's performance on natural language inference tasks is an indication of how well it understands language and even compare it to the famous Turing test which was introduced as *the imitation game* by Turing [1950] and is used to assess whether or not machine can "think". In the current climate of immense popularity of *large language models*, natural language inference is very relevant as these large language models are often judged on their perceived ability to reason logically or think.

In natural language inference the essential semantic concepts *entailment* and *contradiction* are used to describe the relation between texts (Fyodorov et al. [2000], Condoravdi et al. [2003]). The texts, often singular sentences, usually consist of a premise sentence and a hypothesis sentence. The task of the classification algorithm is then to label a pair of premise and hypothesis with one of the following *entailment labels*:

- Entailment, the hypothesis logically follows from the premise.
- Contradiction, the hypothesis contradicts the premise.
- Neutral, the hypothesis does not logically follow from the premise, but it also does not contradict it.

Examples of premise-hypothesis pairs for these relations can be found in Table 1.1. There have been numerous approaches proposed to tackle this classification problem. Bowman et al. [2015b] use a Tree Structured RNN, Chen et al. [2017] an Enhanced LSTM and Radford et al. [2018] use a Transformer Model. Some of these score as high as 89% on the SNLI dataset (Bowman et al. [2015a]), a dataset commonly used in this field of research.

Building on the concept of natural language inference, Xie et al. [2019] introduce the idea of *visual entailment*. In visual entailment, the challenge is labeling a premise image paired with a hypothesis sentence. This still uses the same entailment labels as the natural language inference task, only replacing the premise sentence by an image. Algorithms for this problem have been proposed by Xie et al. [2019], accompanying their introduction of the problem. Cao et al. [2023] use an alignment-based architecture, reaching an accuracy of 72.45% on SNLI-VE.

This SNLI-VE dataset was also introduced by Bowman et al. [2015a], alongside the concept of visual entailment and their approach to solve it. A real-world example of the necessity of well performing visual entailment models is given by Yanaka et al. [2023]. They use a visual

[1] They refer to it as recognizing textual entailment instead of natural language inference.

Premise	Hypothesis	Label
A soccer game with multiple males playing.	Some men are playing a sport.	Entailment
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	Neutral
A man inspects the uniform of a figure in some East Asian country.	The man is sleeping	Contradiction

Table 1.1: One example for each of the three entailment relations.

entailment model for identifying brain tumors in MRI scans. They also introduce a specific medical visual entailment dataset called MedVTE.

For visual entailment models to be trained effectively, availability of datasets is paramount. This work will focus on creating visual entailment datasets, specifically using *generative AI*. The capabilities of generative AI are currently rapidly evolving. Conventional visual entailment dataset creation is often based on human labelling of image-hypothesis pairs using crowdsourcing. Generative AI could reduce cost of dataset generation and save humans from the tedious task of labeling images. This would enable dataset generation for domain specific visual entailment tasks instead of the current general purpose visual entailment that the SNLI-VE dataset contains examples for.

In this thesis we will research *The viability of using generative models for visual entailment dataset creation*. This topic will be divided into research questions which are:

- *How does a synthetic visual entailment dataset compare to an existing visual entailment dataset consisting of real images?*
- *To what extent will a model trained on generated data have similar generalization performance compared to a model trained on original data?*

The main contribution of this work is an AI generated synthetic version of SNLI-VE as well as a synthetic version of Iokawa et al. [2024]'s SICK-VTE. Further contribution is answering the question of whether or not using generative AI is a viable method for creating visual entailment datasets. We verify the validity of the generated images by using intrinsic comparison to the original versions of the generated dataset. This is done by calculating the feature vectors of the original images and checking for the most similar feature vectors among the generated images. We also evaluate the utility of the generated data by training a visual entailment model on our synthetic SNLI-VE dataset and testing it on the original SNLI-VE test set achieving an accuracy of 68.8% compared to the 70.3% accuracy the model trained on real data achieved. We finally briefly look into transfer learning from both generated and original SNLI-VE towards SICK-VTE, also both generated and original.

The structure of the remainder of this thesis is first, we give a more in depth explanation of the background of visual entailment and related work in the field in Chapter 2. We then explore the original SNLI-VE and SICK-VTE datasets and how they were created in Chapter 3. After that we explain our methodology, both for generating the datasets as well as verifying their validity in Chapter 4. The results of the experiments we perform verifying the generated data are shown and explained in Chapter 5. And finally, in Chapter 6 and Chapter 7 we will look into what could have been done differently or in future work and answer the research questions respectively.

Chapter 2

Related Work

This thesis tries to tackle the challenge of dataset creation for visual entailment tasks. In this chapter we take a look at other work on (visual) entailment and creating datasets for these tasks. Note that most of the related work cited in this chapter also performs experiments on the datasets they created, however, in this chapter, the focus lies on the datasets they create.

2.1 Visual entailment and dataset creation

The idea of visual entailment was first proposed by Xie et al. [2019]. For this task they introduce the EVE Model, which is short for Explainable Visual Entailment, and which is explainable by showing where it find the most information. This visualization method is called Attention Visualization. In this paper they also introduce the SNLI-VE dataset which is explained in more detail in Section 3.1.

The idea for visual entailment builds on *visual question answering* and textual entailment. Antol et al. [2015] introduced a dataset for visual question answering. They used the Microsoft Common Objects in Context (MS COCO) dataset by Lin et al. [2014] as a starting point of ∼200k images of real-world scenes with 5 captions per image. To this starting point they added 50k images of abstract scenes for which they also collected 5 captions per image. They used the same user interface for collecting the captions of the abstract scenes as Lin et al. [2014] used for their captions. They experimented with multiple different user interfaces to collect questions based on the images and their captions. This proved difficult as the questions needed to be answerable without any common sense knowledge, only using what is visible in the scene.

Entailment is formalized by Condoravdi et al. [2003] who state that "*the detection of entailment and contradiction relations between texts is a minimal metric for the evaluation of text understanding systems.*"

A dataset for textual entailment is introduced by Marelli et al. [2014] who created the SICK dataset. SICK is short for *sentences involving compositional knowledge* and contains sentence pairs with both relatedness scores and entailment labels. This dataset was created by using the Flickr8K dataset from Hodosh et al. [2013] and the SemEval-2012 STS data from Agirre et al. [2012]. They created the SICK dataset by normalizing, expanding and pairing of sentences in the aforementioned datasets and having Amazon Mechanical Turks workers annotate them with both similarity scores and entailment labels. This took the workers 3 months at a cost of 2030 dollars for the ∼10k sentence pairs in the SICK dataset.

The SICK dataset was later translated to multiple languages, examples of which are Dutch (SICK-NL) and Japanese (JSICK) by Wijnholds and Moortgat [2021] and Yanaka and Mineshima [2022] respectively. Wijnholds and Moortgat [2021] tried to combat the lack of a Dutch NLI dataset to reduce the possibility in NLP research caused by having only English

datasets. They created the Dutch version of SICK using a semi-automatic translation similar to Real et al. [2018]. First, they translated each sentence to Dutch using a machine translator. Secondly, they adapted the translations focusing on meaning being preserved. This step was only necessary for a subset of the machine-translated sentences. Finally, they post-processed the translated sentences to ensure unique translations for unique sentences. They call this alignment.

Yanaka and Mineshima [2022] took a different approach than Wijnholds and Moortgat [2021] in order to get the Japanese translated version of the SICK dataset. They first had an expert translator translate each of the sentences from English to Japanese. The translated sentences were then validated by English-Japanese bilinguals to ensure their correctness. The translator received specific instruction to try to maintain word order where that was possible as they say this is not always natural when translating English sentences to Japanese.

Bowman et al. [2015a] introduced textual entailment dataset: the SNLI dataset on which the aforementioned SNLI-VE was based. Bowman et al. [2015a] claims the SICK dataset is too small and not balanced enough. Balanced implies here that the different entailment labels should occur the same number of times. Their SNLI dataset addresses these problems by creating a balanced dataset of around ∼500k sentence pairs compared to the ∼10k in the SICK dataset. The ∼500k examples in SNLI are split equally between the entailment labels whereas the ∼10k data points in SICK were split into 29% entailment, 57% netural and 14% contradiction.

There are also efforts made to improve existing datasets. This was already the case with Goyal et al. [2017] which improved and extended the VQA dataset resulting in the VQA-v 2 dataset. The dataset was improved by, among other things, reducing bias and extended it by adding more images. This has also been done for the SNLI-VE dataset by Do et al. [2021] who created the e-SNLI-VE. They first corrected what they deemed to be errors in the validation and test sets of SNLI-VE which were too many neutral labels. This resulted in what they called SNLI-VE-2.0. After that they added the human written natural language explanations which were added to SNLI by Camburu et al. [2018] to form e-SNLI to their SNLI-VE-2.0 which resulted in the e-SNLI-VE dataset with improved labels and explanations.

2.2 Synthetic data

Unlike the largely human made datasets that were previously discussed, the following dataset by Johnson et al. [2016] is automatically generated. This dataset is called CLEVR and contains images of abstract shapes combined with automatically generated questions. The images were created by randomly sampling a scene graph and rendering it using Blender[1]. Johnson et al. [2016] use a complex system for automatically generating the questions and answers in which they first choose a question template from which, what they call, a question family. In this template they randomly choose values for the parameters it has. One of the main challenges they face when generating the questions is getting questions that are, what they call, *ill-posed* or *degenerate*. They use the following sentence to explain these terms: "The question *What color is the cube to the right of the sphere?* would be ill-posed if there were many cubes right of the sphere, or degenerate if there were only one cube in the scene since the reference to the sphere would then be unnecessary." As most of the randomly generated questions would fall in one of these categories, they employ a depth-first search to find valid parameters for the question generation algorithm.

There is also research being done towards validation of synthetic data. Livieris et al. [2024] propose an evaluation framework for synthetic data generation models which focuses numerical

[1]Blender is open source 3D rendering software. (https://www.blender.org/)

data which is not really relevant in this thesis. Yuan et al. [2024] created an evaluation framework for assessing synthetic data generated by large language models. This framework focuses on fidelity, utility and privacy. Fidelity focuses on how well the synthetic data resembles the original data, utility is determined by the effectiveness when using the data in downstream machine learning tasks and privacy is important to prevent real sensitive data from being revealed in synthetic datasets. In this work, we only focus on the fidelity and utility of the generated data. The fidelity is tested by comparing the generated images to the original images in a process we call intrinsic comparison whereas utility is tested by training MLP classifiers on the generated data and comparing its performance to an MLP trained on real data.

There are some objections to using generated training data. Some research suggests using synthetic datasets for model training could have a negative effect on performance in the future. Guo et al. [2024] focus on text data where they show a decrease in linguistic richness and variety when language models are trained on synthetic data. They fine-tuned language models on output of previous language models and repeated this process to get the previously mentioned diminishing diversity.

Another paper about possible negative effects of synthetic training data is Hataya et al. [2023]. This paper investigates the effect of using generated datasets for training computer vision models. They conclude that the effect of using generated datasets is negative on the performance of computer vision models. This is very interesting as the subject of this thesis is using generated data to create training datasets.

Both of the just mentioned papers that are worried about the negative impact of generated data focus on a feedback loop. The problem with this feedback loop is that generative models generate new training data on which another (or even the same) generative model is then trained. They suggest this would happen as most training data for generative models is scraped from the internet where there is no real way to check if the scraped images are not generated.

For the image generation in this work this problem is not relevant. The images that we generate are not used as training data for generative models but rather, to train classification models. Furthermore, these classification models are tested on original data, ensuring good real world generalizability and making it irrelevant if the training data is generated or not.

Similarly to the image generation in this work, Askari et al. [2023] also use generative AI to generate training data for other models in a way where this feedback loop effect is not an issue. In their case, the model generating the data is a large language model, as textual data is needed instead of image data. They present a training dataset for cross-encoder re-rankers consisting of generated documents which means that this data should not be used to train generative models, eliminating the feedback loop effect.

Chapter 3

Data

In this chapter the different datasets that were used in this thesis will be explained. Each of these datasets will be given it's own section in which more details are given with regard to the properties of the set, as well as some examples.

The first dataset that will be explained is Xie et al. [2019]'s SNLI-VE. This set will be explained in Section 3.1. After this the SICK dataset, from Marelli et al. [2014], will be explained in Section 3.2.2.

3.1 SNLI-VE

The first dataset that was used for this research was the SNLI-VE dataset which was introduced by Xie et al. [2019] as was mentioned in Chapter 2. This dataset was constructed by combining the SNLI dataset (Bowman et al. [2015a]) from with the Flickr30k dataset (Young et al. [2014]). Each of these datasets are explained in the following subsections. After this it will be clear how the SNLI-VE was created by combining them.

3.1.1 Flickr30k

The Flickr30k dataset was created by Young et al. [2014] by taking 31,783 photos of everyday activities which were harvested from Flickr[1]. They then used crowdsourcing to create captions these images. Each image receives 5 different captions resulting in 158,915 captions in total.

The captions are created by different annotators per image-caption combination, meaning no image will have multiple captions written by the same authors. The annotators are only able to use what is shown in the photos as they have no prior knowledge about them. There is also a different level of specificity used by each annotator which results in different annotations for the same image. Examples given by the authors of the paper for the last two named properties are: "Three people setting up a tent" rather than "Our trip to the Olympic Peninsula", which is an example of how the annotators can only use what is shown without prior knowledge, and "performing a musical piece" vs "bowing on a violin" which are two captions of the same image but using different levels of specificity. In Figure 3.1 an example is shown of an image together with its captions

Their method of generating captions for these images is based on Hodosh et al. [2013]. In Hodosh et al. [2013] they already used this method of crowdsourcing to create a corpus of 8,092 images with 5 captions per image. The Flickr30k dataset is an extension of this corpus and the accompanying papers have some of the same authors.

[1]Flickr (`https://www.flickr.com/`) is a website on which people can share photos.

- A bearded man, and a girl in a red dress are getting married.
- A wedding party walks out of a building.
- The group of people are assembling for a wedding.
- A man and woman dressed for a wedding function.
- A woman holds a man's arm at a formal event.

Figure 3.1: One of the ∼30k photos and its 5 accompanying captions.

3.1.2 SNLI

The SNLI dataset by Bowman et al. [2015a] is a dataset specifically created for natural language inference tasks. The abbreviation SNLI is short for Stanford Natural Language Inference. The dataset was based on the previously mentioned Flickr30k dataset, however, as this dataset only included image-caption pairs instead of the sentence-sentence pairs (from now on called sentence pairs) needed for inference tasks some alterations were needed.

To get from image-caption pairs to sentence pairs, first the images were removed. This left them 158,915 sentences about everyday activities (recall that the Flickr30k photos were selected to depict everyday activities). These ∼160K captions were then used as premises for the sentence pairs.

Amazon Mechanical Turk workers were used to come up with 3 hypotheses per premise. These three hypotheses had to correspond to the entailment, neutral, and contradiction labels. The workers were not shown the original images and could therefore only use the information in the captions provided by the previous dataset. A plot of the sentence length distribution of both the premises from the Flickr30k dataset as well as the newly added hypotheses is shown in Figure 3.2.

As an example of the hypotheses that were added, we take the captions from Figure 3.1 and show which hypotheses are added to these in List 3.1. Bowman et al. [2015a] also states that there were 4K more sentence pairs added to the corpus from the VisualGenome corpus[2]. Statistics about the resulting dataset are shown in Table 3.1. It is interesting that the 570,152 total sentence pair count does not correspond to $3 * \sim 160K$ however, the paper does not go into detail as to where this discrepancy stems from. We find an explanation for this in Section 3.1

After generating these hypotheses, they took measures to validate the data. About 10% of the sentence pairs were validated by Mechanical Turk workers in the following manner. Sentence pairs were given to 4 workers without the label of the author of the hypothesis. These workers then each gave their own label to these sentence pairs resulting in a total of 5 labels per sentence pair (4 new labels and the label the author gave).

Sentence pairs were considered validated if one of the three labels had at least three occurrences in the label set of five, which they called consensus. Such consensus was not found for

[2]The article linked to `visualgenome.org` but this link is no longer active. The VisualGenome corpus has since been expanded and can be found here: `https://homes.cs.washington.edu/~ranjay/visualgenome/about.html`

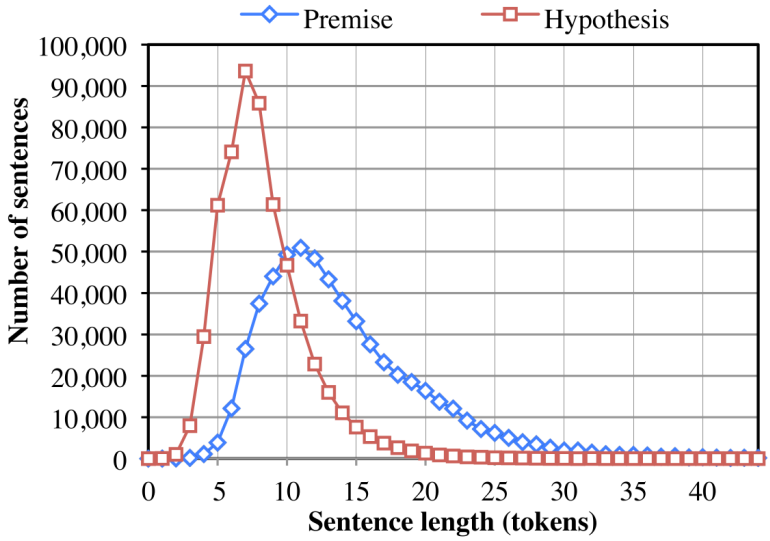


Figure 3.2: The distribution of sentence length. (Figure from Bowman et al. [2015a])

Dataset	Size
Training pairs	550,152
Development pairs	10,000
Test pairs	10,000
Total	570,152

Table 3.1: Basic information about the SNLI dataset.

∼2% of the sentence pairs which were then given a placeholder label. Bowman et al. [2015a] does include these sentence pairs in the dataset but did not use them for any experiments and consider them "*unlikely to be helpful for the standard NLI classification task*".

A problem with this statistic is that, assuming the ∼10% which was validated represents the whole dataset, there are roughly 2% of the non-validated sentence pairs for which validating workers would not find a consensus. Even though this percentage is low, this is something to keep in mind. Another interesting statistic they show is that in only 91.2% the consensus label matches the author label, meaning that in 6.8% percent of the validated sentence pairs, three or more validators agreed on a label that differs from the label the author of the hypothesis intended to be true.

Luckily these problems only hold for small percentages of the dataset, the majority (58.3%) of the sentence pairs were labeled with unanimous consensus of both the author and the validators of the sentence pair. Also, each of the sentence pairs in the test and development set were validated, making these rather trustworthy.

3.1.3 Combining Flickr and SNLI

Now that Young et al. [2014] created a dataset where ∼30K images each received 5 captions and Bowman et al. [2015a] used these captions as premises to create a natural language inference dataset by adding 3 hypotheses per premise, combining the images and captions (premises) from Flickr30k with the hypotheses from SNLI seems an obvious choice to create a visual inference dataset. This is exactly what Xie et al. [2019] did to create the SNLI-VE set.

- A bearded man, and a girl in a red dress are getting married.
    1. The two people are getting married.
    2. The two people are going to live happily ever after.
    3. The beared man and the girl are getting a divorce.
- A wedding party walks out of a building.
    1. the bride and groom walk outside.
    2. the new couple leave the building.
    3. the bride and groom walk into the ocean.
- The group of people are assembling for a wedding.
    1. People are gathering together for a wedding
    2. People are getting together for a friends wedding
    3. The people are running away from a wedding
- A man and woman dressed for a wedding function.
    1. The couple wearing formal wear.
    2. The couple dressed for their daughter's wedding.
    3. The couple wearing swimwear.
- A woman holds a man's arm at a formal event.
    1. She is holding the guy's arm at the event.
    2. She is holding the guy's leg at the event.
    3. She is running away from the guy.

List 3.1: Premises with their hypotheses ordered 1: entailment, 2: neutral, 3: contradiction.

They did make some minor alterations to the datasets they combined to make them more viable for the visual entailment task. Firstly, they removed the sentence pairs for which no consensus was found and were deemed "*unlikely to be helpful for the standard NLI classification task*" by Bowman et al. [2015a]. Secondly, they changed the train-dev-test split from the split that was proposed in SNLI. SNLI has a split where each of the captions from Flickr30k will only be in one of the three sets. This is an obvious choice for a train-dev-test split. Xie et al. [2019] adds the constraint that for every caption, all captions with the same image have to be in the the same part of the dataset. This makes sense as this removes the possibility for an algorithm trained on the training set to have already "seen" an image in the test set.

Some basic statistics of the SNLI-VE dataset are shown in Table 3.2. In this table the number of hypotheses corresponds to the number of sentence pairs as each hypothesis has exactly one corresponding premise. We see that some of the captions are left out as the number of premises is almost but not exactly equal to 5 times the number of images. It is also interesting to note how there are premises with more than 3 hypotheses in this dataset, this follows from the fact that the number of hypotheses, or sentence pairs, is greater than 3 times the number of premises. This solves the question posed in Section 3.1.2 as to where the extra sentence pairs

	Number of images	Number of premises	Number of hypotheses
Partition			
Train	29,783	147,648	529,527
Development	1000	4,960	17,858
Test	1000	4,959	17,901
Total	31,783	157,567	565,286

Table 3.2: Basic information about the SNLI-VE dataset.

in the SNLI set come from.

Further comparing Table 3.2 to Table 3.1 it is clear that some examples are left out. When looking at the total number of sentence pairs, the number goes down from 570,152 in SNLI to 565,286 in SNLI-VE.

From this section we can conclude that the SNLI-VE dataset is large, as we see in Table 3.2, balanced, which follows from the fact that the SNLI dataset was already balanced, and visual, as the Flickr30k dataset was added to have pictures for the premises. That is all information we need about the SNLI-VE dataset.

3.2 SICK-VTE

The second dataset that was used for this research was the SICK-VTE dataset from Iokawa et al. [2024]. SICK-VTE is an abbreviation for Sentences Involving Compositional Knowledge Visual Textual Entailment. This dataset was based on the SICK dataset from Marelli et al. [2014] as was already mentioned in Chapter 2. It also includes additions from SICK-NL (Wijnholds and Moortgat [2021]) and JSICK (Yanaka and Mineshima [2022]) which are the Dutch and Japanese translations of the SICK dataset. In the following subsections we will look at how this dataset was created and how it was based on previous datasets.

3.2.1 Flickr8k

Not unlike the SNLI-VE dataset, it again starts with image data from Flickr. Here the Flickr8k dataset was the starting point instead of the Flickr30k. The Flickr8k dataset was presented in Hodosh et al. [2013]. In Section 3.1.1 it was mentioned how this dataset formed the basis for the Flickr30k dataset.

As this is the basis of the Flickr30k dataset, it shares many of the features of that dataset. It consists of 8,092 images from the Flickr website. As was briefly mentioned in Section 3.1.1, crowdsourcing was used to generate captions for these images. For each of the ∼8k images, 5 captions were created resulting in 40,455 captions[3]. As the Flickr30k dataset was an extension of the Flickr8k dataset, the image-caption pairs of the Flickr8k dataset are also present in the Flickr30k making the Flickr8k dataset a subset.

3.2.2 SICK

The next step towards the SICK-VTE dataset was the creating of the SICK dataset by Marelli et al. [2014]. This dataset was based on the textual part of the Flickr8k dataset combined with another dataset called SemEval 2012 STS MSR-Video Description data set Agirre et al. [2012].

[3] $8{,}092 * 5 = 40{,}460 \approx 40{,}455$

The latter of these will not be explained further in this thesis as its additions are not used in the SICK-VTE dataset which is what was used for the experiments of this work.

The creation of the SICK dataset differs fundamentally from the creation of the SNLI dataset. Where the SNLI dataset was created by giving Amazon Mechanical Turk workers more open instructions of coming up with 3 hypotheses per premise, one corresponding to entailment, one for contradiction and one neutral hypothesis, in the SICK dataset the rules for creating the hypotheses were more strictly defined. First each original sentence was normalized using normalization rules. These normalization rules are shown in Table 3.3. In this table, each S0 is an original sentence and each S1 is its normalized form.

Rule	**Example**
Replace possessive pronouns with the word they stand for or with a determiner.	S0: A man is standing outside his house S1: A man is standing outside the house
Replace Named Entities with a word that stands for the class.	S0: A woman is playing Mozart S1: A woman is playing classical music
In order to avoid generic sentences , transform all non-stative verb tenses into present continuous.	S0: Birds land on clothes lines S1: Birds are landing on clothes lines
Replace complex verb constructions into simpler ones.	S0: A man is attempting to surf down a hill made of sand S1: A man is surfing down a hill made of sand
Simplify verb phrases with modals and auxiliaries.	S0: A kid has to eat a vegetable soup S1: A kid is eating a vegetable soup
Replace phrasal verbs with a synonym if verb and preposition are not adjacent.	S0: A man is sorting the documents out S1: A man is organizing the documents
Remove multi word expressions.	S0: A person is playing guitar right now S1: A person is playing guitar
Remove dates and numbers; if the number is a determiner write it in letters.	S0: 3 people are on a small boat enjoying the view S1: Three people are on a small boat enjoying the view
Turn subordinates into coordinates.	S0: A faucet is running while a bird is standing in the sink below S1: A faucet is running and a bird is standing in the sink below
Turn non-sentential descriptions into sentences.	S0: An airplane in the air S1: An airplane is flying in the air
Remove indirect interrogative and parenthetical phrases.	We did not find any instance in the data sets

Table 3.3: Normalization rules. (Table from Marelli et al. [2014])

After the sentences were normalized, the normalized versions were expanded using a set of expansion rules, creating new sentences in the process. A subset of these expansion rules is shown in Table 3.4. In this table, each S1 is again the normalized version of the original sentence. S2 is created by in a way that the meaning of the S1 sentence should be preserved whereas the S3 and S4 sentences are created by rules that alter the meaning of the S1 sentence. The S3 expansion rules are designed to form negative transformations of S1 whereas the S4 expansion rules are designed to produce sentences with a different meaning using the same

words. Note that Table 3.4 does not contain all expansion rules but at least one rule for each of the three expanded categories (S2, S3 and S4).

Meaning Preserving Transformations **Rule**	**Example**
Turn active sentences into passive sentences and vice versa.	S1: A man is driving a car S2: The car is being driven by a man
Replace words with near synonyms or similar words.	S1: A young boy is jumping into water S2: A young kid is jumping into water S1: A man and two women in a darkened room are sitting at a table with candles S2: A man and two women in a dark room are sitting at a table with candles
Meaning Altering Transformations **Rule**	**Example**
Insert or remove negations to produce contradictions.	S1: The boy is playing the piano S3: The boy is not playing the piano
Scramble words: switch the arguments of a transitive verb, switch and mix modifiers, exploit verb transitive/intransitive alternations, exploit homonymy and polysemy.	S1: The turtle is following the fish S4: The fish is following the turtle S1: A man with a jersey is dunking the ball at a basketball game S4: The game of basketball consists of a ball being dunked by a man with a jersey

Table 3.4: A subset of the expansion rules. (Part of table from Marelli et al. [2014])

From the lists of rules and examples in Table 3.3 and Table 3.4 it is clear that the annotators were not given much freedom in creating the sentences for the SICK dataset. The inclusion of Table 3.3 serves to give the reader an idea of how extensive the rules given to the annotators were. The original version of Table 3.4 in Marelli et al. [2014] is even larger but the main idea of the sentence generation is clear from this simplified version.

In the original SICK dataset, the next step was to create sentence pairs. Which were then judged on relatedness and checked for entailment. Each normalized S1 sentence was paired with all of the expanded sentences that were based on it. Furthermore, random combinations of unrelated sentences were added to the list of pairs as well. This process will not be explained further in this thesis as it is not used for the generation of the SICK-VTE dataset which will become clear in the next section.

3.2.3 Combining Flickr and SICK

In the previous two sections we explained how the Flickr8k dataset was created by using crowd-sourcing to generate captions corresponding to the images as well as how these captions were then normalized and expanded to create the SICK dataset. Comparable to how the SNLI-VE dataset was created by reintroducing the images of the Flickr30k dataset to the textual SNLI dataset, the SICK-VTE dataset was created by combining the Flickr8k images to the SICK sentences.

In the SNLI-VE dataset, the captions of each image were used exclusively as premises in the SNLI dataset, meaning that when the images are reintroduced it is trivial to create premise-hypothesis pairs with images for the premise. This is not the case for the SICK dataset as the captions are used both as premise and hypothesis in different pairs. This results in a situation where just adding the images in the place of their captions each of the pairs, we would end up with premise-hypothesis pairs where the hypothesis is an image. Although we will briefly touch on this idea in Chapter 6, it is not what we need for the experiments in this thesis. To create a dataset that contains images only in the premise part of the pairs, the SICK-VTE dataset uses the premise image and an expanded sentence based on a caption of that image along with its expansion rule. This results in image-hypothesis-rule combinations where this corresponds to an entailment label[4].

The resulting SICK-VTE dataset only contains entailment and contradiction labels (no neutral) which is a result of the pairs being created by the expansion rules which either preserve or alter the meaning of the sentence as was shown in Table 3.4. In the original SICK dataset random samples of unrelated sentences are also included which result in neutral pairs. Some basic statistics of the SICK-VTE dataset can be found in Table 3.5.

From this table it is clear that the SICK-VTE dataset is much smaller than the SNLI-VE dataset. It is also clear that the SICK-VTE dataset is not balanced in terms of number of entailment, contradiction and of course neutral examples. Recall how these were two of the reasons for the creation of the SNLI-VE dataset.

Number of unique images	**Number of unique captions**	**Number of pairs**	**Entailment examples**	**Neutral examples**	**Contradiction examples**
488	1,909	2,899	1,930	0	969

Table 3.5: Basic information about the SICK-VTE dataset.

[4]The SICK-VTE dataset also contains entailment labels which, without exception, correspond to the rule for each of the combinations.

Chapter 4

Methods

The methods chapter of this work consists of two main parts. In the first part we will discuss the image generation for each of the datasets and explain which models were used and why. In the second part we explain how we verify that the generated images can be used as training data for visual entailment models.

4.1 Image generation

In this section we explain how the image generation is performed. First we provide a short explanation on the model that was used. After that, we give information about the parameters that used in the model. Finally we discuss what was generated for the different datasets that were introduced in Chapter 3. This chapter also includes introductory information about verifying the validity of the resulting images which will be given in Section 4.2.

4.1.1 Image generation model

There are many different models that can be used for image generation nowadays and with the current interest in generative AI this number is also increasing. OpenAI has DALL-E[1] as a text-to-image generative AI model. Another example of a generative AI model is Midjourney[2] which is hosted and created by Midjourney, Inc. a San Francisco based research lab.

Although the aforementioned generative models create stunning images, they are not open source and not able to be run on a local machine. That is why, for this thesis, the choice of generative model is Stability AI's[3] Stable Diffusion[4].

Stable Diffusion is an open source text-to-image generative AI model which can be run on local computers[5]. The ability to run the model locally as opposed to the cloud based solutions from OpenAI and Midjourney was essential for generating the large amount of images necessary for this research. This model being open source also has the added benefit that there are multiple finetuned versions of it for different purposes. Many of these finetuned models can be found on Huggingface[6] where they can be freely downloaded to experiment with.

The Stable Diffusion model was based on a project called Latent Diffusion which was introduced in Robin et al. [2022]. In this paper the inner workings of the Stable/Latent Diffusion

[1]DALL-E website: https://labs.openai.com/
[2]Midjourney website: https://www.midjourney.com/
[3]Stability AI website: https://stability.ai/stable-image
[4]Stable Diffusion GitHub page: https://github.com/Stability-AI/generative-models
[5]The local computer should preferably have a GPU with enough ($\geq$ 8GB) of VRAM, but running it on the CPU is also possible with greatly reduced performance.
[6]Huggingface website: https://huggingface.co/

model is explained in detail. In short, Diffusion Models generate images by starting with an image of random noise and iteratively denoising that image towards the desired output. This is done by, in each time step, predicting the noise that should be removed and removing that. The main contribution of Robin et al. [2022] is departing from pixel space towards latent space. Latent space provides a great reduction in the dimensionality compared to pixel space which greatly increases performance. This is just a high level explanation of Latent Diffusion on which Stable Diffusion is based. For a more in depth explanation the reader is refereed to the Latent Diffusion paper.

Robin et al. [2022] also mention how both training and inference of a diffusion model is very computationally expensive. They measure the computational cost of training a model in V100 days, named after the NVIDIA V100 GPU commonly used for machine learning tasks, which goes down from 150–1000 V100 days for conventional diffusion models to just 35 V100 days for their Latent Diffusion model. Even though this is a huge performance improvement, 35 V100 days is still an enormous computational cost. This is why the pre-trained and finetuned models on Huggingface are such a useful resource. It is hard to really put into perspective how high the computational cost of training such a model is. In Appendix A a better attempt will be made by comparing V100 days to the hardware used for this thesis but for now, we give a quick comparison from the V100 datasheet[7], the performance of 1 V100 GPU equals 32 CPUs meaning 35 V100 days would be ∼1,000 CPU days.

4.1.2 Model parameters

In this section we give an overview of the choices made when choosing the parameters of the generative model. Firstly we take a look at the resolution of the generated images. The chosen resolution was square images of 512x512 pixels. This choice was based on the following factors:

- The most important reason to choose the resolution of 512x512 pixels for the generated images is the fact that the Stable Diffusion model was trained on images of this size[8] as it used a subset of the LAION-5B[9] dataset. Note that in the bar chart shown in Figure 4.1 we use binning to group together the occurrence counts. This might give the impression that images with a height of 300 to 350 are the most common this is an effect of that binning. The most occurring value for both the height and width is 500 pixels however, the most of the images have a non-square aspect ratio with the height being smaller than the width.

- 512x512 is also close to the most common image resolution in the original SNLI-VE dataset where the most common value for both width and height is 500. There are, in that dataset, no images larger than 500 pixels in either widht or height but, given that the Stable Diffusion model performs best at 512x512 pixels resolution, 512 was considered close enough to 500 for this research. In Figure 4.1 we show a chart of the distribution of the images, both for width and height.

- The final reason for choosing 512x512 pixels as the size for the generated images is that experiments with smaller images yielded very unclear images as a result in which it was hard for a human to see what was depicted. Generating larger images resulted in very slow performance. 256x256 was tried as a smaller size which did speed up the generation

[7]Datasheet: https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf

[8]Stable Diffusion HuggingFace page: https://huggingface.co/blog/stable_diffusion

[9]LAION-5B website: https://laion.ai/blog/laion-5b/

but visual inspection showed that the images lacked detail and 1024x1024 was tried as the larger size but this made the generation process about 4 times slower than 512x512 sized images.

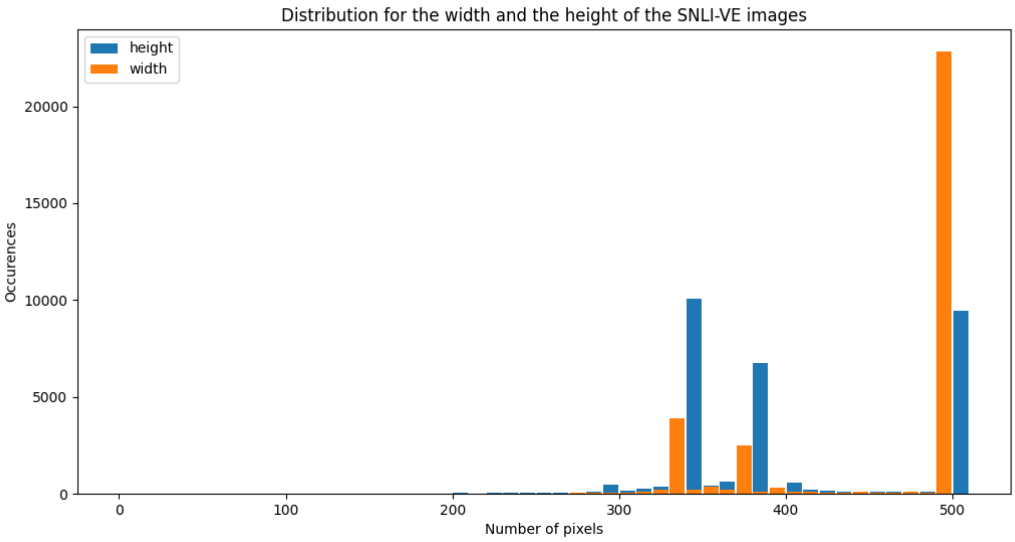


Figure 4.1: Distribution of the widths and heights of the images in the SNLI-VE dataset.

Another important parameter choice for this research was which Stable Diffusion checkpoint that was used. As mentioned in Section 4.1.1, Stable Diffusion has a variety of finetuned versions. The versions are designed for different purposes with some being more aimed at artistic image generation (e.g. DreamShaper[10]) whereas the ToonYou[11] checkpoint generates cartoon-like images. The checkpoint chosen for this research is the focused on generating photorealistic images. This seemed most fitting when the original SNLI-VE images are in fact photographs. The checkpoint we therefore used is the RealisticVision checkpoint[12]. This is a popular checkpoint for generating photorealistic images rather than being artistic or cartoon-like.

4.1.3 Datasets

In this section we discuss what was generated for the different datasets. First we dive into the SNLI-VE dataset and it's generated version. After that we take a look at what was generated for the SICK-VTE dataset.

Generated SNLI-VE

The first dataset that was generated for this thesis was a synthetic version of the SNLI-VE dataset. Recall from Chapter 3 how the original SNLI-VE dataset consists of ∼32k images. Also recall how each of these images is accompanied by around 5 captions (premises).

For the generated version of this dataset, one image was generated for each of these captions. This results in 157,567 generated images in total. These generated images represent a visual version of what the annotators found most relevant in the original image as different annotators

[10]DreamShaper HuggingFace page: https://huggingface.co/Lykon/DreamShaper
[11]ToonYou HuggingFace page: https://huggingface.co/stablediffusionapi/toonyou
[12]RealisticVision HuggingFace page: https://huggingface.co/stablediffusionapi/realistic-vision-v51

A wedding party walks out of a building.

The group of people are assembling for a wedding.

A man and woman dressed for a wedding function.

A woman holds a man's arm at a formal event.

Figure 4.2: Four out of the five images that were generated for the captions in Figure 3.1.

focused on different parts of the original image when writing captions and these captions form the basis of the generated images. This results in images that are different from each other and the source image, showing some overlap in semantic content but focusing on different aspects of the original just as the annotators had done. In Figure 4.2 we see four examples of different images generated for 1 source image in the original SNLI-VE dataset. This source image together with its captions was introduced in Chapter 3 in Figure 3.1. The four images in Figure 4.2 are missing one version that was generated from the missing caption. This image is shown in Figure 4.3.

The missing example in Figure 3.1 has intentionally been left out of that set of images as it contains an interesting example. The caption "*A bearded man, and a girl in a red dress are getting married.*" could be considered ambiguous in who is (or are) wearing a dress. The generative model seems to have had difficulty interpreting this ambiguous caption as the resulting image, shown in Figure 4.3, shows both the man and the woman who are getting married, wearing dresses. This is of course not what the annotator meant when describing the original image and would not be what most humans envision when reading the annotator's caption, however it is interesting to see how a generated image can differ from the original image for this reason.

The ambiguity of the captions might be a problem for more of the generated images. This

A bearded man, and a girl in a red dress are getting married.

Figure 4.3: Language ambiguity example.

is not something that is easily checked as there are 157,567 generated images in this dataset alone. The example in Figure 4.3 was found when randomly sampling some of the generated images to check if they looked somewhat realistic.

One ambiguous example of course doesn't say much about the validity of the generated images as a whole. An explanation on how a more thorough investigation on the generated images was performed experimentally is provided in Section 4.2. The results of these experiments are presented in Chapter 5.

Generated SICK-VTE

As introduced in Section 3.2.2, the other dataset used for this thesis is the SICK-VTE dataset. We also created a synthetic version of this dataset. This was done similarly to the synthetic SNLI-VE dataset that was explained in the previous section which is why this section will be less extensive, mainly focusing on the differences and other remarkable parts of this generation process.

Recall from Table 3.5 that the original SICK-VTE dataset contains 488 unique images with which 2,899 pairs are formed. What is also shown in that table is the number of unique captions for these images that are used as the original sentences from which the hypotheses are formed using the expansion rules. Recall how this number is 1,909. This becomes relevant here as these sentences are used to generate the images for the synthetic SICK-VTE dataset. This means that for most of the pairs in this generated dataset, a unique image is used. This is unlike the synthetic SNLI-VE dataset as for every image in that set, almost always, 3 hypotheses are included.

Another difference from the generated SNLI-VE dataset is that the generated SICK-VTE dataset does not include neutral pairs. This naturally follows from the fact that the original SICK-VTE dataset does not include any of these pairs, as was explained in Section 3.2.2. This is important to note however, as this means that this dataset can not be used as a training set for visual entailment models. In Chapter 6 we investigate the idea of generating neutral examples for this set as well, but for the experiments in this thesis it is important that the format is similar to the original SICK-VTE dataset. How this is the case as well as which experiments this synthetic dataset is used for will both be explained in Section 4.2.3.

4.1.4 Challenges

There are a couple of challenges that emerged with the generation of the images for the previously named datasets. These challenges apply to both the generation of the synthetic SNLI-VE dataset as well as the synthetic SICK-VTE dataset.

The first of these challenges includes the generation of hands and fingers. This is a well known problem in multiple image generation models. Recently interesting new approaches have been proposed to solve this shortcoming of generative AI however, in the model that was used in the experiments in this thesis, these new approaches are not implemented yet

Yang et al. [2024] have introduced a framework for the training of generative models that specifically aims to improve the generation of hands and fingers. Narasimhaswamy et al. [2024] created a model named HanDiffuser which is, as the name implies, a diffusion model that performs better when images of hands are required.

For this work the problem of generation of hands is not specifically an issue as the original images of the datasets used are aimed to depict everyday activities. The sentences that are written as captions for these images and the hypotheses that accompany them usually do not contain specific information about the hands and fingers of the people performing these activities. However, if a visual entailment dataset is needed to train a model to recognise how many finger someone shows or which hand gestures a person does, using generative AI to create this set would be a challenge.

One similar issue that might pose a problem is the image in the bottom right corner of Figure 4.2. Here we see a man with an odd looking hand holding a woman's arm whereas the caption stated *"A woman holds a man's arm at a formal event."* This role reversal might pose problems in an entailment dataset as such an inversion of action is what have been used as an expansion rule when creating the SICK-VTE dataset.

Another challenge for in the generation of this dataset was the high inference cost of the Latent Diffusion model. This is also mention in Robin et al. [2022] and was mainly a problem for the generation of the synthetic SNLI-VE dataset.

As the SNLI-VE dataset contained more than 150k images, inference cost was a real concern here. On an NVIDIA GTX1080 GPU, which is the GPU used for this thesis, generating one image takes roughly 40 seconds. $157{,}567 * \sim 40$ seconds $\approx$ 6,302,680 seconds $\approx$ 1,751 hours $\approx$ 72 days. This was certainly a challenge but this was reduced by using multiple GPUs. More information on the hardware used in this work can be found in Appendix A. In Chapter 6 the challenge of computation cost will be explained with a focus on the possibility of future work building on this thesis.

As a result of the computational costs being very high, the choice of parameters and checkpoints for the generation model was also challenging. Using multiple checkpoints and other parameters was not feasible due to the long runtime of the generation experiment. The possibility of doing this in the future will again be looked into in Chapter 6.

4.2 Image verification

We test the validity of the generated images using a variety of experiments . These experiments are broadly categorised into three groups: *intrinsic comparison*, *classification* and *transfer learning*. Each of these groups gets a subsection in this section in which we explain the setup of the experiments. The results of these experiments will then be shown in Chapter 5. Not all experiments were performed on all datasets as some experiment-dataset combinations do not make sense. In the following sections we will go into detail as to why that is the case.

4.2.1 Intrinsic comparison

For the first of the three experiment categories compared the generated images to the original ones. We call this intrinsic comparison as the intrinsic similarity of features of the images. This experiment was performed on the SNLI-VE dataset combined with its generated version. Recall from Section 4.1 how each generated image was based on a premise, which was in turn based on a caption from an original image. Each generated image therefore has a parent image from which it stems and as each original image has around five different captions, it also has around five different child images. An example of a parent and child image is provided in Figure 4.4, The other child images based on this parent image[13] were already shown in Figure 4.2 and Figure 4.3.

A man and woman dressed for a wedding function.

Figure 4.4: An example of a parent (left) and a child (right) image.

These child images can be compared to the images they originated from and checked to see if they are similar. However, this similarity is hard to quantify which is why we need a measure to judge the similarity as well as a way to quantify it. Similarity scores are also included in the SICK dataset (Marelli et al. [2014]) where the similarity between sentences was judged by crowdsourcing, however for the problem of measuring similarity between the original and generated pictures here, such a method is not feasible due to the large number of

[13] One might call these siblings.

potential images and the extensive process of comparing each possible pair. The SICK dataset only contains ∼10k pairs where only the similarity between the sentences in each pair were evaluated, whereas this problem problem has $n * m$ number of comparisons where n is the number of original images and m is the number of generated images. Recall how n and m are ∼30k and ∼160k respectively.

As crowdsourcing clearly is not the solution here, we calculated the similarity measure by using CLIP (Radford et al. [2021]) to generate feature vectors from both the original images and the generated images after which the similarity between these vectors was calculated by using the cosine similarity score. These feature vectors represent intrinsic features of the images, this is why this experiment is called intrinsic comparison.

The similarity scores of the feature vectors is just a number without unit which makes it hard to draw conclusions from. This problem was tackled by judging the similarity by using the *recall@k* score. For each original image we compare it to all generated images and sort the list of generated images by similarity score. We then check how high the corresponding generated child images are ranked in this list. The recall@k measure is then calculated based on this ordered list where each generated image that is a child image of the original is considered to be a true positive and each generated image that is not a child image of the original is considered a false positive. We take the average recall@k score over all of the original images which gives an indication on how similar the generated child images are compared to the original data.

More information about the specific experiments in this class, their parameters and their results will be given in Section 5.1. Here we also explain some practical problems that were faced with this form of experimentation.

4.2.2 Classification

For the next class of experiments the generated images were used as training data for a classifier to learn the visual entailment classification problem. The approach for this experiment was based on Song et al. [2022] who proposed using CLIP for visual entailment.

Their method includes taking the CLIP feature vector of both the premise image and the hypothesis text, fusing these using a `fuse` function 4.1 and training an MLP on this fused vector representation to output the correct entailment label.

$$fuse(v1, v2) = [v1, v2, v1 + v2, v1 - v2, v1 \cdot v2] \tag{4.1}$$

This method was also used for this thesis where separate classifiers were trained on both the both the original images and the generated images of the SNLI-VE dataset. These classifiers were then tested on both the original and the generated test sets as well after which their performance was compared.

For this thesis the absolute performance of the classifier is not the main goal. We are more interested in the relative performance of a classifier trained on generated images compared to a classifier trained on real images. We however aim for good performance of both as this yields the most accurate data to compare between these two.

More information about the parameters of these experiments, e.g. the shape of the network, the optimizer used and the learning rate, is presented in Section 5.2. Here we also look at the performance of the classifiers and other interesting results from these experiments.

4.2.3 Transfer learning

For the final group of experiments in this thesis we look at transfer learning. In this part we see how well the trained models perform on an entirely different dataset. For this we use the

trained models from Section 4.2.2 and check how they perform on the SICK-VTE dataset. For completeness we also check the performance of these trained models on the synthetic version of SICK-VTE that was generated.

Transfer learning experiments from SNLI to SICK have already been performed with varying results. Talman and Chatzikyriakidis [2019] suggest that transfer learning from SNLI to SICK results in poor performance whereas Bowman et al. [2015a] seem to be more optimistic about the generalization ability. However, both of these look at textual inference whereas in this thesis we investigate the performance on visual entailment, where we again do not focus on the absolute performance of the classifiers but focus more on the relative performance of the model trained on real data compared to the model trained on generated data.

As was the case in the previous subsections of this section, the results of these experiments are presented in a different part of this thesis. For this group of experiments the details, results and parameters will be explained in Section 5.3. Here we will also explain why the focus is on transfer from SNLI-VE to SICK instead of vice versa as well as other choices that were made when designing these experiments.

Chapter 5

Experiments and Results

In this chapter we go into more detail about the experiments that were introduced in Section 4.2. The experiments are explained, the results are shown and the results are interpreted in this chapter. After reading this chapter, the reader should have a clear understanding about what was done in this thesis to check the validity of the generated images and investigate the viability of using generative AI for visual entailment dataset creation.

As was the case in the previous chapter, each of the experiments is given it's own section, starting with the intrinsic comparison (Section 5.1), followed by the classification experiment (Section 5.2) and finally the transfer learning experiment (Section 5.3)

5.1 intrinsic comparison

For the first experiment the generated images were compared to the original images to which they correspond. Recall from Section 4.2.1 how we coined the terms parent image and child image, where the parent image is one of the original images from the dataset and its children are the generated images based on one of the parents accompanying captions. It was also already introduced that the comparison measure between the images is the cosine similarity of the feature vectors of the images, where the feature vectors are generated using CLIP. These cosine similarities are ordered to where the generated images most similar to the original should be highest in this list.

In the following subsections we find out if the similarity of generated child images and their parents is indeed high. We introduce and use a sampling method to make the problem of finding the similarities more feasible. We also look at the total distribution of similarity scores. And we show some counter example as well as some other challenges to show that this similarity score is not perfect for judging the generated images.

5.1.1 Similarity distribution

Before getting into the judging of the generated images using the similarity to their parent images, we first want to get an idea of the similarity scores as a whole. As mentioned before, we defined the similarity between images as the cosine similarity between their feature vectors. In this subsection we take a look at the distribution of this similarity between the original and the generated images.

First we take a look at the similarity distribution for images in the the development and test set of the SNLI-VE dataset and its generated child images. Each original image is compared to all the generated images and the similarity scores are saved. Figure 5.1 shows a bar chart

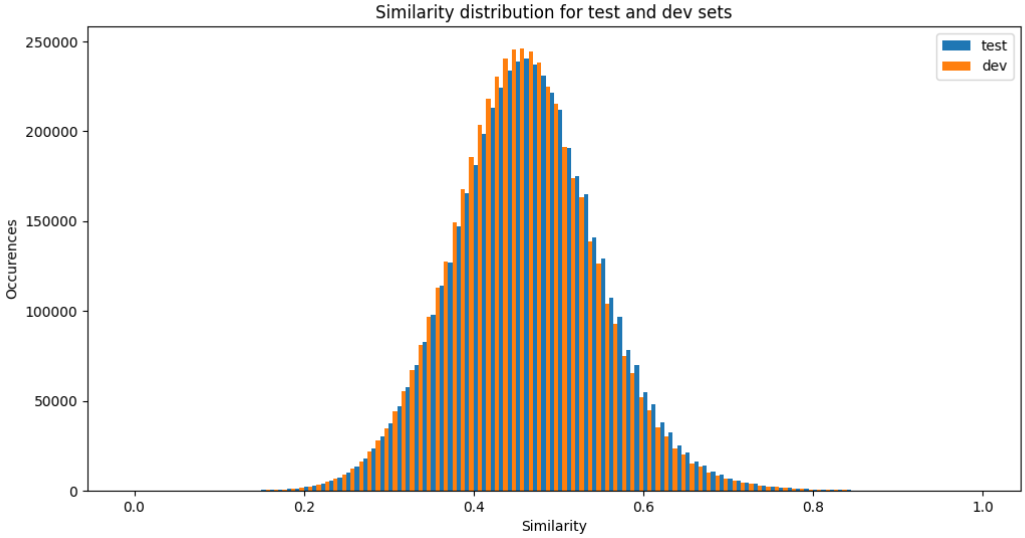


Figure 5.1: Similarity distribution for the development and test set.

	Mean	**Standard deviation**	**Toal number of similarity scores**
Partition			
Development	0.4652	0.0845	4,959,000
Test	0.4655	0.0861	4,956,000
Full dataset	0.4648	0.0860	5,007,824,829

Table 5.1: Means, standard deviations and number of similarity scores.

for these similarities. We used 100 bins to plot the continuous distribution of cosine similarity values which obviously range from 0 to 1.

From the bar chart it clearly follows that the similarity values follow a normal distribution for the development and test set. The means and standard deviations for these partitions of the dataset can be found in Table 5.1.

A chart, similar to the one in Figure 5.1, of the distribution of the similarity scores of the full dataset is shown in Figure 5.2. This distribution is very similar to the distribution of the development and test set and Table 5.1 shows its mean and standard deviation to be very similar as well.

Note that information about the distribution of the similarity scores for the train set without the test and development sets is omitted for the charts and table. This is because of the high computational cost of running these similarity experiments combined with the fact that the results for the full dataset and the development and test partitions is very similar. As mentioned in Section 4.2.1, the complexity of getting the similarity scores for the datasets is $n * m$ where n is the number of original images and m the number of generated images that are candidates to be similar. This results in a very large number of calculations for larger dataset which is also shown in Table 5.1. For the development and test set, this number is quite manageable as $n = 1000$ and $m \approx 5 * n$ resulting in $n * m \approx 1000 * 5000$ for these sets.

For the full dataset, as well as the train set that was omitted, this number is way higher. Here n and m are ~30k and ~160k respectively resulting in the total number of similarity calculations

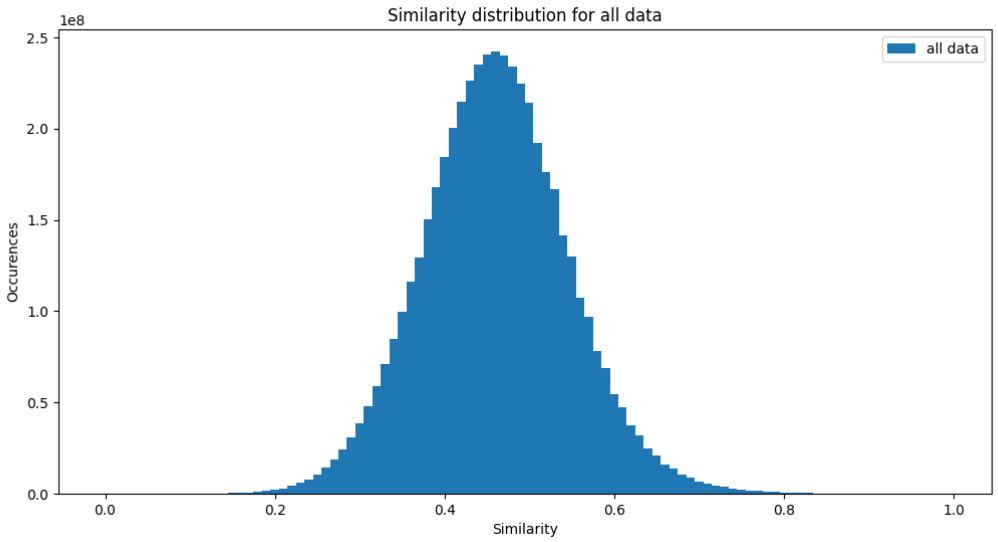


Figure 5.2: Similarity distribution for the full dataset.

being in the order of billions. Running the experiment on the whole dataset took multiple days and repeating it for the train set alone[1] would not add much interesting information which is why it is omitted.

Now that we have an indication of how the similarity scores are distributed we can look at how the actual results look for the experiments based on these scores. This is be done in the following subsections.

5.1.2 Ranked similarity

In this section we look at the ranked similarity scores of the full dataset. As was introduced in Section 4.2.1, we specifically look at the recall@k score. First we explain this measure as well as its counterpart precision@k.

The recall@k measure is often used as a ranking metric for judging document retrieval based on a certain query Lin et al. [2020]. In Lin et al. [2020] recall is defined as "fraction of relevant documents (in the entire collection C) for q that are retrieved in ranked list R", or:

$$\text{Recall}(R, q) = \frac{\sum_{(i,d)\in R} \text{rel}(q, d)}{\sum_{d\in C} \text{rel}(q, d)} \tag{5.1}$$

Where R is the ranked list, q is the query, $rel(q, d)$ is the relevance score of a document d given query q, i iterates over the length of the retrieved list and C is the entire collection of documents. To adapt this to the problem of finding and comparing the relevant generated images, we change this equation slightly, partly out of necessity and partly to make it more intuitive for this problem.

In the ranking problem in this work, the relevance function is binary where it returns 1 for an image that was indeed a child image given a certain original image as a query and 0 if it is not. The query is the original image and the ranked list consists of the 100 most cosine similar generated images that were found and the collection of documents is the set of generated images. As almost every original image has 5 child images that would be considered relevant,

[1]The train set "alone" implies: whole dataset \ (test set ∪ dev set)

the bottom part of the equation is omitted. This also results in the recall value no longer being between 0 and 1 but rather between 0 and 5 which we considered more intuitive when working with a fixed number of relevant results more query. The resulting equation for recall in this specific takes the following form:

$$\text{Recall}(R, q) = \sum_{i=1}^{100} \text{rel}(q, R[i]) \tag{5.2}$$

Here, R is ranked list of generated images and $R[i]$ is the image on position i in this list. The rest of the variables are basically unchanged.

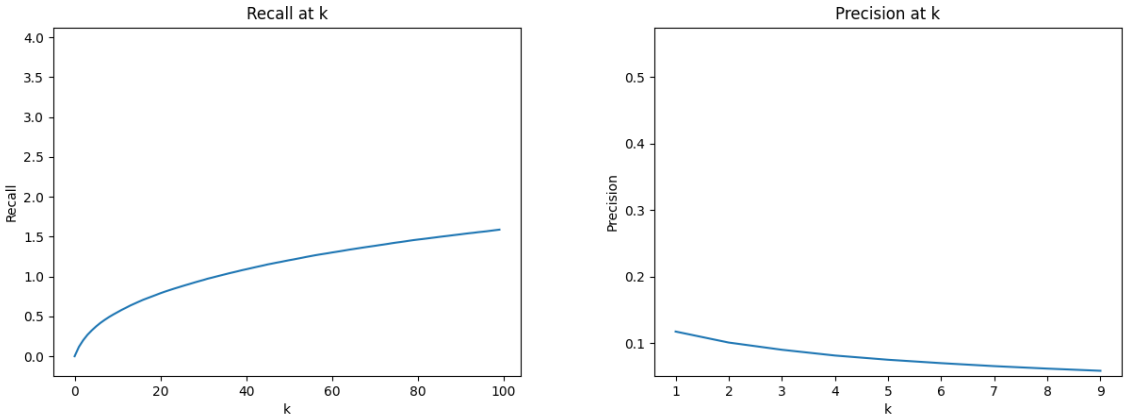
Recall@k is easily defined from the aforementioned equation by not looking at the recall for the whole list of length 100 but looking at the first k elements. This results in the following equation:

$$\text{Recall@k}(R, q, k) = \sum_{i=1}^{k} \text{rel}(q, R[i]) \tag{5.3}$$

For the precision@k measure we start at the recall@k equation above and modify it to take into account the number of retrieved images. This results in the following equation:

$$\text{Precision@k}(R, q, k) = \frac{\sum_{i=1}^{k} \text{rel}(q, R[i])}{k} \tag{5.4}$$

To get an idea how well the generated images resemble the original images, we plot the recall@k and precision@k curves. For this, the recall@k and precision@k values were averaged over all query images. The resulting figures for the recall@k and precision@k can be found in Figure 5.3a and Figure 5.3b respectively.



(a) (b)

Figure 5.3: Recall (a) and precision (b) curves.

Interpreting these curves we find that in the 100 most similar images according to cosine similarity, there are only 1.6 images that are considered relevant with relevance defined as being a generated image using one of the captions from the query image as generation prompt. The precision curve is also rather low and is capped way before $k = 100$ as, otherwise, the resulting curve was not visible anymore.

These results stem from the fact that finding the 100 most similar out of ∼160k generated images will likely not result in finding all of the 5 images that are relevant. In the next section

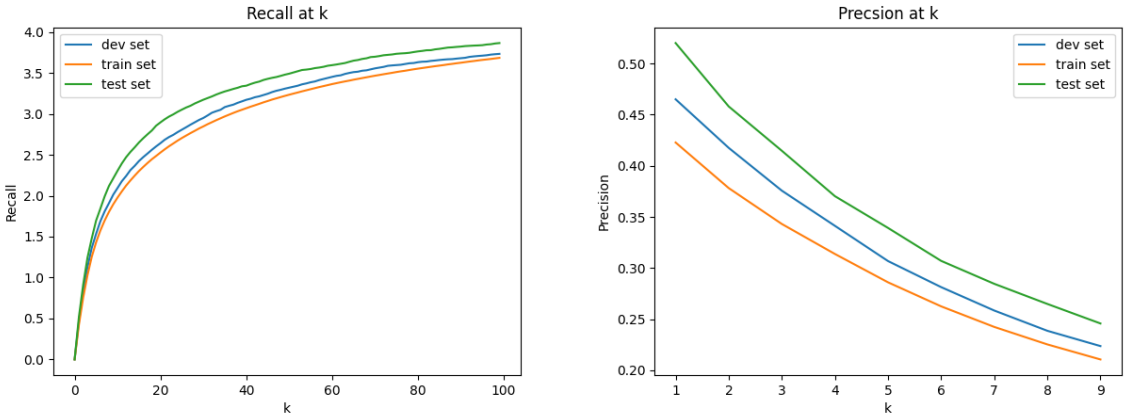
we use sampling to tackle this problem and in Section 5.1.4 we go into more detail about the different challenged faced in this experiment.

5.1.3 Sampled ranked similarity

As mentioned in the previous section, here we look at the same experiment but using samples rather than the whole dataset. For the sampling we first used the test and development set as defined in the SNLI-VE dataset. For these samples we conducted the same experiment as before, taking the 100 most similar generated images for every query image and ranking these images. After this the same precision@k and recall@k curves are plotted.

To see if the train set is similar to the development and test sets we sampled it to the same size as those sets. Recall how both the test set and the development set contain 1000 query images and their generated counterparts contain ∼5k resulting images. As the size of the train set is not divisible by 1000, some of the samples overlap resulting in some query images and their generated counterparts being in multiple samples. This was necessary as the size of the sample of queries dictates the size of the collection of possible results as only the generated images that are based on the query images in the sample are considered when calculating the similarity scores.

The resulting plots for the recall@k and precision@k of the samples can be found in Figure 5.4b and Figure 5.4a respectively. Note that the train set curve is the average of all the samples in the train set whereas the development set curve and the test set curve are only based on their respective sets.

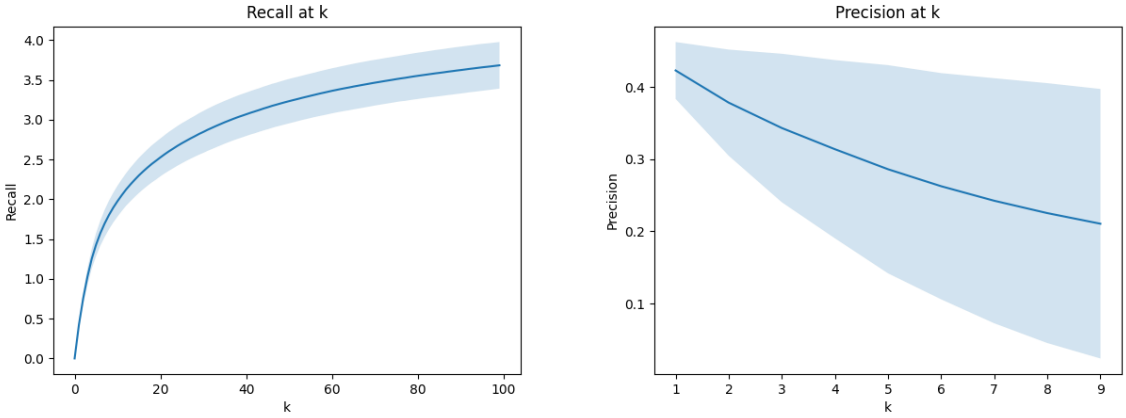


(a) (b)

Figure 5.4: Sampled precision and recall curves.

From these figures we see that, when using this sampling, most of the relevant images are found within the first 100 most similar generated images. For each of the different sets the average result is between 3.5 and 4 out of the five possible relevant images. The precision curve is still very low and therefore capped just as before. The precision curve mainly serves as a zoomed in view of the start of the recall curve as that curve is not really readable in the first values of k.

Finally for completeness we now take a look at the variance of the recall and precision curves of the samples. These curves are shown in the subfigures of Figure 5.5 where one standard deviation above and below each curve is marked. From these plots where the curves and standard deviations are visualized we see that the variance is not very large. However, the



(a) (b)

Figure 5.5: Average precision and recall curves with one standard deviation.

precision plot does seem to show quite a large variance.

5.1.4 Challenges

In this subsection we look at some of the challenges that were faced in this experiment. Firstly, we discuss the problem with large dataset sizes as was encountered in Section 5.1.2. The computational cost of of comparing all of the images is very high and scales quadratically with the number of inputs.[2] This makes that it would not scale very well for use with even larger datasets than what was used in this thesis.

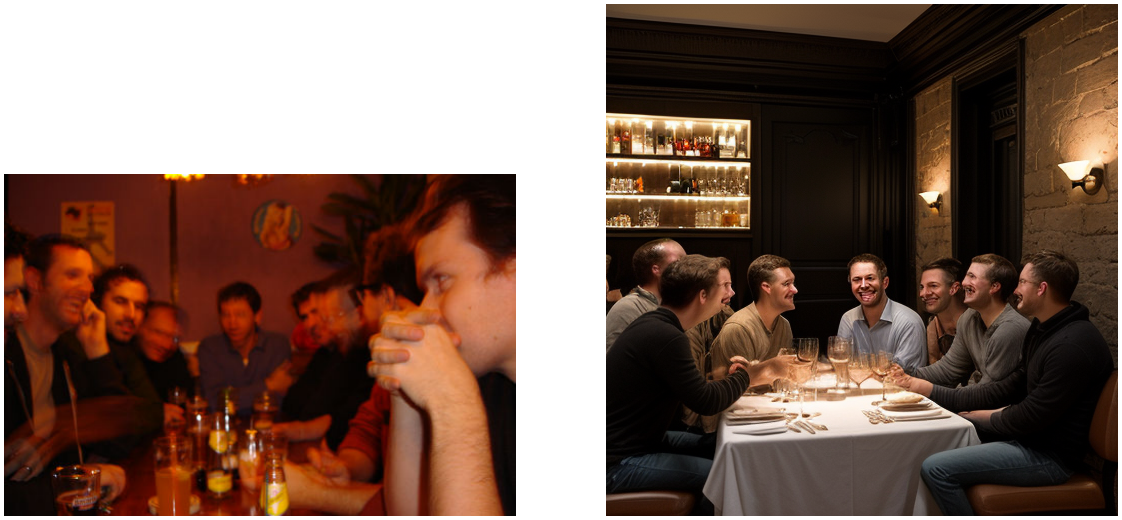
Secondly, similar to the computational cost, the storage cost for all similarity scores is also immense. This is why for the experiment in Section 5.1.2, only the first 100 were considered even though setting this retrieved image number higher would result in more representative scores.

Thirdly, the relevance function is not without flaws. The relevance here is defined only by looking at whether the image was generated from one of the captions of the query image or not, however other images can be very similar to the query image but not considered as such. An example of this is shown in Figure 5.6 where an image is shown together with the most cosine similar generated image which is not one of its child images. These two images could be considered rather similar by a human. It is likely that there are more images in the collection that are similar than only the child images, making the recall@k measure an underestimation of the real quality of the generated images.

5.2 Classification

The second part of the experiments in this thesis include training a classifier to correctly label the premise images and hypothesis pairs to an entailment label. Section 4.2.2 already introduced roughly how this was done and how it is based on Song et al. [2022]. In this section we take a more in depth look at how the classification experiments were designed as well as the results they yielded. First we explain what kind of classifier was used, after that we explain the experimental setup in more depth and finally we show the results of the experiments.

[2] Actually $O(n * m)$ but this is quadratic as m is correlates linearly with n.



(a) Original (b) Generated

Figure 5.6: An example of an image and a generated image which looks similar but is not considered relevant as the generated image is not a child of the original image in this evaluation.

5.2.1 Classification model

For the model to classify the image-hypothesis pairs we used a multilayer perceptron. This is in accordance with Song et al. [2022] who also took this approach. The input dimension for this perceptron is 2560 which is a direct result of the output size of the $fuse$ function which was already shown in Function 4.1 but, as a reminder, can also be found in Function 5.5. Recall from Section 4.2.2 that this is the same function used in Song et al. [2022].

$$fuse(v1, v2) = [v1, v2, v1 + v2, v1 - v2, v1 \cdot v2] \tag{5.5}$$

The $fuse$ function concatenates the feature vector of the image, the feature vector of the hypothesis, the sum of these two vectors as well as the difference between these vectors and finally the product of these vectors. This results in a total of five vectors that are concatenated and with each vector having a size of 512 numbers, the result has a length of $5 * 512 = 2560$.

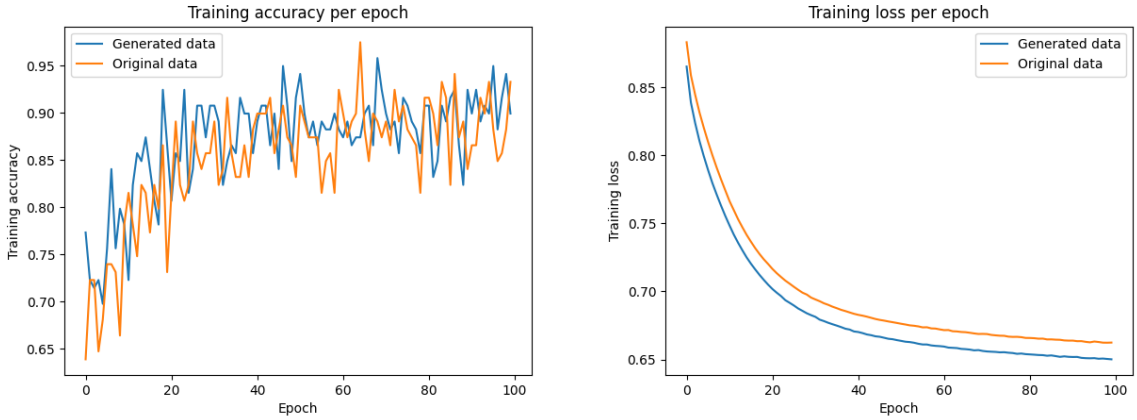
The resulting vector is used as an input for the MLP which has one hidden layer of size 250. The size of this hidden layer did not seem to affect the accuracy of the classifier very much but had an impact on the computational performance. After this one hidden layer the network only has one more layer which is the output layer. This output layer has a size of 3 corresponding to the three possible labels: entailment, neutral, contradiction.

5.2.2 Experimental setup

In order to test the performance of a classifier trained on generated images we had to create such a model. This model could then be compared to a similar model trained which was trained on the original images. Both of these models are of the same basic parameters as outlined in Section 5.2.1. Both were trained on the SNLI-VE dataset, one on the original dataset and the other on the generated version. The same train-test-dev split is used as is suggested by the SNLI-VE dataset.

To get the best performing trained models for both of the datasets we focused on generalization ability. This was done by training the model on the training set for 100 epochs and checking its performance on the development set after each epoch. The model with the best

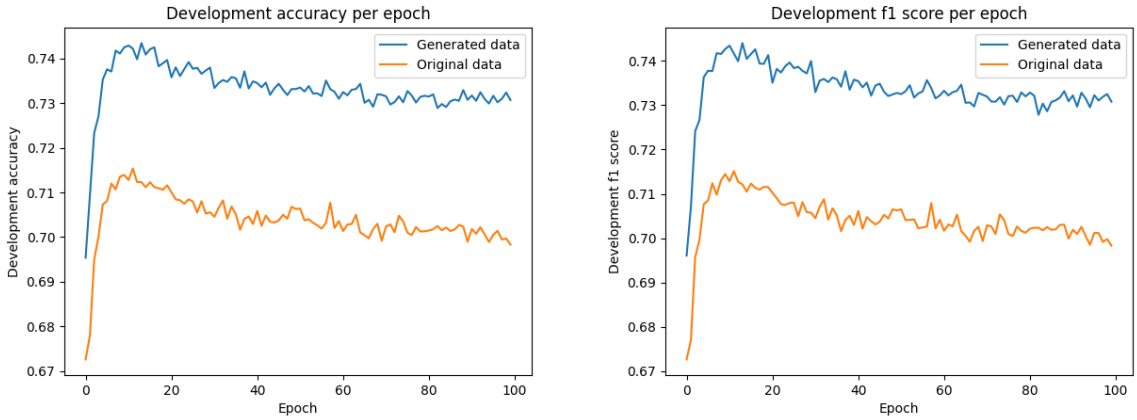
performance on the development set was saved to use in later testing. In Figure 5.7 plots of the training loss (Figure 5.7b) and accuracy (Figure 5.7a) are presented which show that the performance on the training set increases during the training epochs.



(a) (b)

Figure 5.7: Performance on the training set during training.

However, as was already stated, we focus more on the performance of the models on the performance on the development set to ensure better generalizability of the trained models. In Figure 5.8 plots of the accuracy on the development set (Figure 5.8a) and F1 scores (Figure 5.8a) are presented. The models are evaluated on these measures after every training epoch where the model with the highest development accuracy is kept as the final model.



(a) (b)

Figure 5.8: Performance on the development set after each epoch.

From the plots in Figure 5.8 and the plot in Figure 5.7b we can clearly see that the model trained on the generated images achieves a better performance than the model trained on original data when it is also evaluated on the same kind of data. In Section 5.2.3 we see what the performance is on the test set for these models where the model trained on generated data is also evaluated on the original data and vice versa. We also see from the plots in Figure 5.8 that 100 epochs is more than enough as the curves of the performance on the development set suggest for both models that overfitting starts to occur before 20 epochs. As the best performing

Train set \ Test set	Original	Generated
Original	70.3%	71.1%
Generated	68.9%	73.2%

Table 5.2: Accuracies of both models on both test sets.

Train set \ Test set	Original	Generated
Original	0.703	0.710
Generated	0.686	0.732

Table 5.3: F1 scores of both models on both test sets.

models on the development set are the ones that are eventually tested this is not a problem, however using fewer training epoch would have saved time and computational resources.

5.2.3 Testing and results

In this subsection we look at the performance of the trained models explained in Section 5.2.2. As briefly mentioned before we look at the performance of both models on both datasets. We show a matrix for the accuracies in Table 5.2 and one for the F1 scores in Table 5.3.

From the results in these tables we can draw multiple conclusions. First of all, we see that the F1 scores and the accuracy values are very similar for all results. This is caused by the dataset being almost perfectly balanced. Secondly, we observe the best overall performance when using the model trained on generated data evaluated on the generated data as well. This suggests that the generated images and their classification has less variability compared to the original data and thus, more easily generalizes to the test set. We can draw the same conclusion from the better training performance of the model trained on generated data. Thirdly, and perhaps more interestingly, we see that the model trained on original images performs better on the generated test set than it does on the original test set. This is an interesting and unexpected result which could suggest that the generated test set is "easier" to classify. Lastly, we do see that the model trained on generated data and tested on original data has the worst performance in this experiment. Although the performance is similar, it is still lower than all of the other results which suggests that generated training data results in slightly worse performance in real world tasks.

5.3 Transfer learning

The final part of the experiments in this thesis look at the performance of the trained models when they are tested on another dataset. The idea is based on transfer learning where the "knowledge" the models have learned on the SNLI-VE dataset is tested on a different dataset, in this case the SICK-VTE dataset. The experiments in this section are very similar to the experiments in the previous section which is why we focus more on the differences in dataset and results as opposed to the experimental setup.

5.3.1 SICK-VTE as a visual entailment dataset

First we discuss the usefulness of SICK-VTE as a test set. As discussed in Chapter 3 and repeated in Chapter 4, SICK-VTE and its synthetic counterpart do not contain any neutral

Train set \ SICK-VTE	Original	Generated
Original	50.7%	51.4%
Generated	47.2%	47.6%

Table 5.4: Accuracies of both models on the SICK-VTE datasets.

examples. This makes it less desirable as a dataset for visual entailment. To train visual entailment models, having neutral examples would be essential however for the purpose of testing the generalizability pretrained models, a dataset with neutral examples would merely be preferred.

Because of this the choice was made to only test the models that were trained on the SNLI-VE in the previous experiment on the SICK-VTE dataset and not the other way around. In Chapter 6 the possibility of using the SICK-VTE dataset to train models is explored.

5.3.2 Experimental setup

As was mentioned before, the experimental setup is very similar to the experimental setup in Section 5.2.2. The difference here is that the training and validation set are from one dataset and the test set is from a different set. Also, as the models were already trained in the previous experiment, this experiment had no training part.

In order to test the viability of using synthetic data for model training, the generalizability of the model trained on synthetic SNLI-VE is compared to the performance of the model trained on the original SNLI-VE dataset. The training part of the models is done in the previous experiment and the best performing models on the development set were tested on the SICK-VTE dataset. Both of the trained models were tested on the original SICK-VTE dataset and, for completeness, also on the generated version of SICK-VTE.

5.3.3 Results

For the results of this experiment we look at the performance metrics of the models trained on SNLI-VE when tested on SICK-VTE. As was the case in Section 5.2.3, the metrics on which the models' performance is tested are accuracy and F1 score. A matrix containing these metrics can be found in Table 5.4 and Table 5.5 respectively. The same format is used as in Table 5.2 and Table 5.3 to make comparison easier.

From these tables we draw multiple conclusions. First of all, the accuracy of $\sim$50% is about as good as a coin flip as the only classes in the test set are entailment and contradiction and getting half of those correct is the same as random chance. When realizing that a naive baseline of classifying all examples as entailment would get an accuracy score of $1{,}930/(1{,}930 + 969) = 1,930/2{,}899 = 0.6657$, which is about two thirds, a score of just over 0.5 is even worse.

This result suggests that transferring of visual entailment knowledge from the SNLI-VE dataset to the SICK-VTE dataset does not work for the models trained in this work. Recall how Talman and Chatzikyriakidis [2019] found similar performance issues when doing the same for textual entailment models trained on the SNLI dataset and tested on the SICK dataset.

Secondly we can conclude that the model trained on generated data performs slightly worse compared to the model trained on original data. This is in line with the findings in the previous experiment where the models were tested on the test set of the SNLI-VE dataset. The performance difference between the model trained on original data and the model trained on synthetic data is also similar to before. However, as the performance of all of the model-test

Train set \ **SICK-VTE**	Original	Generated
Original	0.400	0.391
Generated	0.384	0.384

Table 5.5: F1 scores of both models on the SICK-VTE datasets.

set combinations is rather bad, this might be less significant. More research could be done to find whether or not different models could have better performance when transferring from SNLI-VE to SICK-VTE

Chapter 6

Discussion

In this chapter we reflect on the rest of the work and look at what could have been done differently or what can be done in the future, building upon this work. First we take a more high-level look at the contributions of this work, after this we will focus on the limitations of this work and finally we give suggestions for future work. These three topics will be handled in the following three sections respectively.

6.1 Contributions

The contributions of this thesis consist of two parts, first images were generated to form synthetic datasets, after that these generated images were experimentally verified to show their validity. A synthetic version was made for both the SNLI-VE dataset as well as the SICK-VTE dataset. The synthetic SNLI-VE dataset contains 157,567 images with almost all of these images having three associated hypotheses, one for entailment, one for contradiction and one for the neutral label. The synthetic SICK-VTE dataset is considerably smaller than the synthetic SNLI-VE dataset. This dataset contains 1,909 generated images, one for each of the image captions in the original SICK-VTE dataset, forming 2,899 image-sentence pairs. This dataset is also considerably less balanced as 1,930 of these pairs get the label of entailment, 969 get a contradiction label meaning that there are 0 neutral exaples. The generation process for these dataset is explained in Section 4.1.

The verification of the images is done in three experiments. These are introduced in Section 4.2 after which they are explained in more detail in Chapter 5 where their results are also shown. A high level overview of the results is given in the following list:

- In the first part of the verification experiments we investigated the intrinsic similarity of the generated images compared to the original versions. This was done by calculating the feature vectors of both the original and the generated images and ranking the generating images by cosine similarity to the original images. These orderings were made in batches of 1000 original images is the size of the development and test set. Using the measure of recall@k we found that within the first 100 of the most similar generated images, on average, between 3.5 and 4 out of 5 possible images were found. More in depth information on this experiment and its results is given in Section 5.1.

- For the second verification experiment we looked at the utility of the generated data. This was done by training two MLP classifiers, one on the generated SNLI-VE dataset and the other on the original version, and comparing their performance. Both models were tested on both the original test set and the generated version of the test set. On the original test

set, the model trained on the original data scored an accuracy of 70.3% and the model trained on generated data scored an accuracy of 68.9%. On the generated test set, the model trained on original data scored an accuracy of 71.1% whereas the model trained on generated data scored an accuracy of 73.2%. More information on this experiment as well as the F1 scores of the models can be found in Section 5.2.

- For the final part of the verification experiments we looked at the transfer learning ability of the models from the second experiment by testing them on a different dataset. This was done by testing both models using the SICK-VTE dataset as a test set. Both the original and the generated version of the SICK-VTE were used for this. The model trained on the original SNLI-VE dataset got accuracy scores of 50.7% and 51.4% on the original and generated SICK-VTE dataset respectively. The model trained on the generated version got accuracy scores of 47.2% and 47.6% on the original and generated SICK-VTE dataset in the same order. This experiment is explained in more detail in Section 5.3.

6.2 Limitations

Although this work is made with the utmost care it still has some limitations. The most important limitation is the fact that only one set of parameters for the generation algorithm was used to generate a full dataset. This limitations stems from a lack of computational resources making generating multiple versions of the synthetic SNLI-VE dataset infeasible. Specifically using different checkpoints and comparing their outputs could have been very interesting.

Although using multiple configurations of the generation algorithm would have been interesting, we do not think it impacts the results in this work. This is because this work mainly serves as a proof of concept to see if using generated images to train visual entailment classifiers is feasible. Using different checkpoints could be interesting but is not necessary for this purpose.

Building on the previous limitation, there is another limitation with the same cause. For each of the captions of both of the datasets only one image was generated. This still results in 5 times as many images as were in the original datasets as the images were accompanied by 5 different captions, however, generating even more images per caption and checking the performance of the trained algorithms given this larger set of training data would be interesting.

We think generating multiple images and checking if the performance gets better would have added something to this work. A big advantage of using generated training data for algorithms is that the generation of this data is very cheap. Looking into the performance of the algorithms in Chapter 5 but increasing the number of generated images per captions would give an indication of how viable this approach really is. In Section 6.3 we look into this idea a bit more.

The final limitation in this work is the classification algorithm used in Section 4.2.2 and Section 4.2.3. This was implemented as a simple MLP taking the fused feature vectors of the image and the hypothesis as input and outputting an entailment label which was based on the approach used by Song et al. [2022]. Using a different algorithm for the visual entailment task, like EVE from Xie et al. [2019], training that on the generated data and comparing it to its original counterpart could yield interesting results.

As the purpose for this work is mainly a proof of concept for using generated datasets to train visual entailment models, searching for the differences in performance of different models is outside of the scope. Furthermore, not all models are trained as easily as a simple MLP, either due to computational costs or due to the model not being open source.

6.3 Future work

In this section we explain what we think would be interesting additions to this work for future research. First of all we address the limitations explained in the previous section.

- Firstly, the single set of parameters for the generation model. The solution to this problem is trivial as it comes down to either using more (or more capable) hardware or giving it more time. Particularly the choice of checkpoint would be interesting to vary as there are numerous checkpoints for many different purposes[1]. As the field of image generation is rapidly evolving, doing the same experiments on a new algorithm altogether could be interesting research.

- Secondly, solving the problem of generating multiple images per caption is even more trivial than the first limitation. This does not even require different settings, just more time or compute power. It would however, be very interesting when datasets with multiple different images per caption are used to find the correlation between number of images generated per caption and performance of the algorithm that is trained on these images. This could show that more generated images for the same set of captions would improve the performance of the algorithm. If that is the case that would be strong evidence in favor of using generated images for visual entailment dataset creation.

- Lastly, training a different classification algorithm on the generated data and comparing it to models trained on a real dataset. This could even be seen as an automated machine learning problem where the goal is to find the best way of training a classification algorithm using generated data. Smith-Miles [2009] and White et al. [2023] give overviews of approaches for algorithm selection and neural architecture search from an automated machine learning perspective respectively. These ideas could be applied for finding algorithms that generalize well well after being trained on generated data.

Now we propose some future research that is not based on the limitations of this work. Using image generation, we can essentially turn a natural language inference dataset into a visual entailment dataset. A specific use for this technology could be to add neutral examples to the SICK-VTE dataset based on the neutral sentence pairs in the original SICK dataset. Another possible direction for experimentation could be looking into ways to use prompt engineering to generate images with specific attributes to make them more suitable for use in a training dataset. This could include but is not limited to, indicating in the prompt that the role reversal that was mentioned in Section 4.1.4 is not allowed. We do not know whether having the generative model pay extra attention to specifics of the image via prompt engineering would alter the utility of the generated data, however, as prompt engineering has been a topic of interest for many people since the recent popularity of generative AI, this could be a something to investigate.

A more novel idea in the field of entailment/inference would be image-image entailment. In this case, both the premise and the hypothesis would consist of an image where the hypothesis would get an entailment label based on if it semantically follows from what is seen in the first image. This would be a possible use case for image generation as it gives a lot of freedom for the dataset generation. It will however, pose many questions that need to be answered such as:

- *How do we define inference on an image-image basis?*

- *How do we decide which part of the image is the subject/most important?*

[1] A checkpoint could even be finetuned specifically for visual entailment dataset generation.

- *To what level of detail do images have to match to receive the entailment label?*

These questions are fundamental to a possibility of image-image entailment problems and are by no means all questions that need answering on that topic. We do not try to answer any of these questions as it is outside of the scope of this work however, if anyone would attempt image-image entailment dataset creation, we recommend at least looking into using image generation.

Chapter 7

Conclusion

To conclude this thesis, we review and answer the research question and its subquestions based on the results of the experiments from Chapter 5. First we take a look at what was done in this thesis after which we answer the subquestions and draw our conclusion for the research question.

In this thesis we looked at the viability of using generated data for visual entailment dataset creation. This was done by generating synthetic versions of both the SNLI-VE dataset (Bowman et al. [2015a]) and SICK-VTE dataset (Iokawa et al. [2024]). These synthetic datasets were then subjected to a multitude of experiments. First, we looked at intrinsic similarity of the generated images and their original counterparts. After the intrinsic comparison, we trained classifier models on both the original and the synthetic versions of the SNLI-VE dataset for which we compared the performance on the SNLI-VE test set. Finally we looked at the transfer learning capabilities of the models trained on SNLI-VE when tested on SICK-VTE. Now we will look at the research questions and draw the final conclusion of this thesis.

The first subquestion we answer is: *"How does a synthetic visual entailment dataset compare to an existing visual entailment dataset consisting of real images?"* We conclude that the images generated for the synthetic version of the SNLI-VE dataset are similar to the images in the original SNLI-VE dataset. This was shown in Section 5.1 where the generated images were compared to the original images and ranked on similarity after which we calculated the recall.

For the second subquestion *"To what extent will a model trained on generated data have similar generalization performance compared to a model trained on original data?"* we find that the performance of the model trained on synthetic data is slightly worse than the performance of a similar model trained on original data. This follows from the results in both Section 5.2, where the models were trained and tested on the synthetic and original versions of the SNLI-VE dataset, as well as Section 5.3 where the same models were tested on the synthetic and original versions of the SICK-VTE dataset. In both of these sections we found the results of the model trained on original data to slightly outperform the model trained on synthetic data where in Section 5.2 the results of both models were way higher than chance accuracy and in Section 5.3 a random guesser would have performed similarly.

Even though the performance of the model trained on generated data was slightly below the performance of the model trained on original data, we still think an accuracy score of 0.688 on the original SNLI-VE test set is very similar to the accuracy of 0.703 of the model trained on original data. Overall we are positive about *the viability of using generative models for visual entailment dataset creation* which was the main topic of this thesis.

Bibliography

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015. URL http://arxiv.org/abs/1505.00468.

Arian Askari, Mohammad Aliannejadi, Evangelos Kanoulas, and Suzan Verberne. A test collection of synthetic documents for training rankers: Chatgpt vs. human experts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 5311–5315, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615111. URL https://doi.org/10.1145/3583780.3615111.

Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL https://aclanthology.org/H05-1079.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. Recursive neural networks can learn logical semantics. In Alexandre Allauzen, Edward Grefenstette, Karl Moritz Hermann, Hugo Larochelle, and Scott Wen-tau Yih, editors, *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21, Beijing, China, July 2015b. Association for Computational Linguistics. doi: 10.18653/v1/W15-4002. URL https://aclanthology.org/W15-4002.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *CoRR*, abs/1812.01193, 2018. URL http://arxiv.org/abs/1812.01193.

Biwei Cao, Jiuxin Cao, Jie Gui, Jiayun Shen, Bo Liu, Lei He, Yuan Yan Tang, and James Tin-Yau Kwok. Alignve: Visual entailment recognition based on alignment relations. *IEEE Transactions on Multimedia*, 25:7378–7387, 2023. ISSN 1941-0077. doi: 10.1109/tmm.2022.3222118. URL http://dx.doi.org/10.1109/TMM.2022.3222118.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017. doi: 10.18653/v1/p17-1152. URL http://dx.doi.org/10.18653/v1/P17-1152.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45, 2003. URL https://aclanthology.org/W03-0906.

Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. e-snli-ve-2.0: Corrected visual-textual entailment with natural language explanations. *CoRR*, abs/2004.03744, 2021. URL https://arxiv.org/abs/2004.03744.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. A natural logic inference system. 12 2000.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837:6325–6334, 07 2017. doi: 10.1109/CVPR.2017.670. URL https://arxiv.org/abs/1612.00837.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3589–3604, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-naacl.228.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 20498–20508. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01879. URL https://doi.org/10.1109/ICCV51070.2023.01879.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47(1): 853–899, 05 2013. ISSN 1076-9757. doi: 10.1613/jair.3994. URL https://dl.acm.org/doi/10.5555/2566972.2566993.

Nobuyuki Iokawa, Gijs Wijnholds, and Hitomi Yanaka. Multilingual visual-textual entailment benchmark with diverse linguistic phenomena. *Proceedings of the Annual Conference of JSAI*, JSAI2024:4C3GS1104–4C3GS1104, 2024. doi: 10.11517/pjsai.JSAI2024.0_4C3GS1104. URL https://www.jstage.jst.go.jp/article/pjsai/JSAI2024/0/JSAI2024_4C3GS1104/_article/-char/en.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016. URL http://arxiv.org/abs/1612.06890.

Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *CoRR*, abs/2010.06467, 2020. URL https://arxiv.org/abs/2010.06467.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Ioannis E Livieris, Nikos Alimpertis, George Domalis, and Dimitris Tsakalidis. An evaluation framework for synthetic data generation models. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 320–335. Springer, 2024. URL https://arxiv.org/abs/2404.08866.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2468–2479, June 2024. URL https://arxiv.org/abs/2403.01693.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. OpenAI, 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020.

Livy Real, Ana Rodrigues, Andressa Vieira e Silva, Beatriz Albiero, Bruna Thalenberg, Bruno Guide, Cindy Silva, Guilherme de Oliveira Lima, Igor C. S. Câmara, Milos Stanojevic, Rodrigo Souza, and Valeria de Paiva. SICK-BR: A portuguese corpus for inference. In Aline Villavicencio, Viviane P. Moreira, Alberto Abad, Helena de Medeiros Caseli, Pablo Gamallo, Carlos Ramisch, Hugo Gonçalo Oliveira, and Gustavo Henrique Paetzold, editors, *Computational Processing of the Portuguese Language - 13th International Conference, PROPOR 2018, Canela, Brazil, September 24-26, 2018, Proceedings*, volume 11122 of *Lecture Notes in Computer Science*, pages 303–312. Springer, 2018. doi: 10.1007/978-3-319-99722-3_31. URL https://doi.org/10.1007/978-3-319-99722-3_31.

Rombach Robin, Blattmann Andreas, Lorenz Dominik, Esser Patrick, and Ommer Bjorn. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2022. doi: 10.1109/cvpr52688.2022.01042. URL https://cir.nii.ac.jp/crid/1360016870030521856.

Kate A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.*, 41(1), jan 2009. ISSN 0360-0300. doi: 10.1145/1456650.1456656. URL https://doi.org/10.1145/1456650.1456656.

Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.421. URL https://aclanthology.org/2022.acl-long.421.

Aarne Talman and Stergios Chatzikyriakidis. Testing the generalization power of neural network models across NLI benchmarks. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4810. URL https://aclanthology.org/W19-4810.

A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. ISSN 00264423. URL http://www.jstor.org/stable/2251299.

Colin White, Mahmoud Safari, Rhea Sukthanker, Binxin Ru, Thomas Elsken, Arber Zela, Debadeepta Dey, and Frank Hutter. Neural Architecture Search: Insights from 1000 Papers. *arXiv e-prints*, art. arXiv:2301.08727, January 2023. doi: 10.48550/arXiv.2301.08727. URL https://ui.adsabs.harvard.edu/abs/2023arXiv230108727W.

Gijs Wijnholds and Michael Moortgat. SICK-NL: A dataset for Dutch natural language inference. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1474–1479, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.126. URL https://aclanthology.org/2021.eacl-main.126.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706, 2019. URL http://arxiv.org/abs/1901.06706.

Hitomi Yanaka and Koji Mineshima. Compositional evaluation on Japanese textual entailment and similarity. *Transactions of the Association for Computational Linguistics*, 10:1266–1284, 2022. doi: 10.1162/tacl_a_00518. URL https://aclanthology.org/2022.tacl-1.73.

Hitomi Yanaka, Yuta Nakamura, Yuki Chida, and Tomoya Kurosawa. Medical visual textual entailment for numerical understanding of vision-and-language models. In Tristan Naumann, Asma Ben Abacha, Steven Bethard, Kirk Roberts, and Anna Rumshisky, editors, *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 8–18, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.clinicalnlp-1.2. URL https://aclanthology.org/2023.clinicalnlp-1.2.

Yue Yang, Atith N. Gandhi, and Greg Turk. Annotated hands for generative models. *CoRR*, abs/2401.15075, 2024. doi: 10.48550/ARXIV.2401.15075. URL https://doi.org/10.48550/arXiv.2401.15075.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions.

Transactions of the Association for Computational Linguistics, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006.

Yefeng Yuan, Yuhong Liu, and Liang Cheng. A multi-faceted evaluation framework for assessing synthetic data generated by large language models, 2024. URL https://arxiv.org/abs/2404.14445.

Appendix A

Hardware

In this appendix we will discuss the hardware used to generate the images and perform the experiments. Due to the long runtime of the image generation process, running it on university servers was infeasible. Both the image generation as well as the experiments were therefore performed on a personal computer. The specifications of this personal computer are shown in Table A.1.

CPU	RAM	GPU	Storage	Operating system
i7 7700k	32GB	3*NVIDIA GTX1080 8GB	256GB SATA SSD	Fedora Linux 38

Table A.1: Specifications of the computer used for this research.

In Section 4.1 it was cited from Robin et al. [2022] that training a diffusion model on local hardware was infeasible as the computational costs was ∼35 V100 days. To put this into perspective, we compare the NVIDIA V100 GPU to the NVIDIA GTX1080 GPU. According to the V100 datasheet[1] the NVIDIA V100 can perform 130 teraFLOPS[2], whereas the NVIDIA GTX1080 can only perform 9 teraFLOPS[2] according to its whitepaper[3]. This makes the V100 roughly 14.4 times faster than the GTX1080 which makes 35 V100 days roughly equal to $35 * 14.4 \approx 505$ GTX1080 days. This still does not paint the full picture as the GTX1080 has only 8GB of VRAM, compared to up to 32GB for a V100, which might hinder training performance even further.

Section 4.1 also mentioned that inference, generating the images, being computationally expensive. On a single GTX1080 GPU, generating one image according to the parameters used in this work takes about 40 seconds. This makes that generating 157,567 images, using three GPUs, took about $40\text{s} * 157{,}567/3 = 2.1 * 10^6$ seconds $\approx$ 25 days. This is not exactly how long the generation process took in this thesis as at the start of the process, there was only one GTX1080 available. Luckily, as this is an older GPU, there are second hand options for this GPU which roughly cost 100 euros[4] which makes it very affordable compared to the many thousands of euros an NVIDIA V100 costs. This, in combination with open source image generation models, greatly democratizes image generation.

[1]Datasheet: https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf

[3]Whitepaper: https://www.es.ele.tue.nl/~heco/courses/ECA/GPU-papers/GeForce_GTX_1080_Whitepaper_FINAL.pdf

[3]TeraFLOPS is short for 10^{12} floating point operations per second.

[4]This what was paid for the two added GPUs in February 2024.