Generating Clip Art Images from Textual Descriptions using Statistical Machine Translation and Mixed Integer Linear Programming Models

Lukas Dirzys

MInf Project (Part 2) Report

Master of Informatics
School of Informatics
University of Edinburgh

2016

Abstract

In this report we propose a method to generate novel images, coherent with given textual descriptions, using phrase-based Statistical Machine Translation and Mixed Integer Linear Programming models. To focus on the semantic meaning understanding rather than reliable feature extractor, we use a dataset of clip art images instead of real photos.

We build a two step system where the first one is aimed to translate input descriptions into an intermediate representation or so-called visual sentences. These specify the spatial relations between objects discovered in the textual descriptions, and are used as an input in the second stage which renders an image, satisfying visual sentences. This is achieved by setting up a Mixed Integer Linear Program.

We compare our model with three baselines- a strong novel approach by Zitnick et al. (2013) and two straightforward image retrieval methods. We propose an automatic metric, based on Visual Paraphrasing task, for evaluating generated images, and perform a human based analysis to manually confirm the automatic evaluation scores. Our proposed model proved to significantly outperform two image retrieval methods, and was competitive with the remaining strong baseline model.

Acknowledgements

Special thanks to my supervisor, Prof. Mirella Lapata, for proposing an interesting project topic, and for her constant help during the course of the project.

I would also like to thank Carina Silberer for helping me in setting up the Phrase Based Statistical Machine Translation model described in Section 2.2, and installing packages required for this model to be trained.

Contents

Chapter 1

Introduction

Learning the relation between natural language and the corresponding visual depiction remains one of the most challenging areas of research in computer vision. Humans, unlike computers, are able to easily imagine the scene captured by the textual story. They use common sense, imagination, and creativity to visually perceive objects and actions represented in the descriptions. For instance, ten humans imagined the simple given description in Figure 1.1 as shown in the corresponding images. All humans chose the appropriate children positions, their body positions and facial expressions, and imagined the ball to create a scene coherent with the given description. However, due to human imagination, other objects (e.g. sun, plants, animals, playground or wearable items) were added into the scene to make it more elaborate. Consequently, humans are capable of imagining diverse range of semantically meaningful scenes.

Computers, unfortunately, are not yet as evolved in this task. There are many problems such as reliable object recognition, reliable image features extraction, and the discovery of the text-to-visual relations correspondence, which need to be fully solved in order to automatically generate images coherent with the given descriptions.

The automated image generation task is challenging, however, further research can help to better understand human behaviour when imagining scenes, and to relate the information between text and image domains better, and if successfully solved, it could

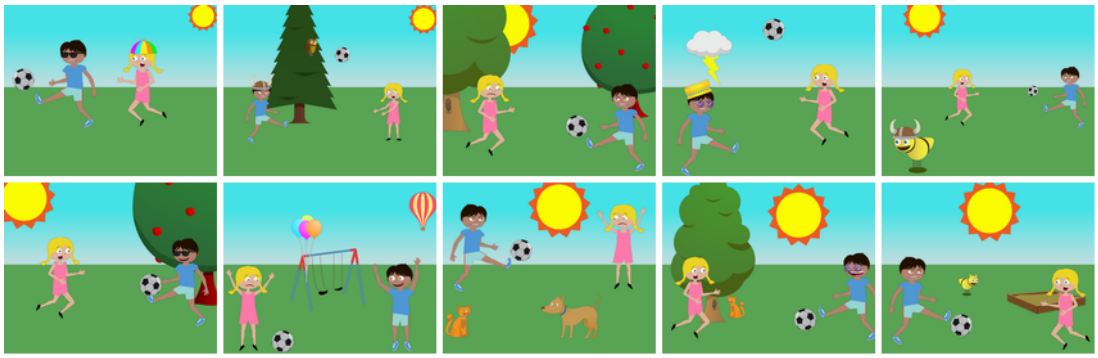


Figure 1.1: Scenes imagined by ten different humans for the description *Jenny loves to play soccer but she is worried that Mike will kick the ball too hard.*

lead to a number of helpful real life applications. For instance, some applications include more descriptive and intuitive sentence-based (instead of keyword-based) image search, automatic 3D game scenes generation, and face visualisations directly through language (Coyne and Sproat (2001) and Perlin and Goldberg (1996) respectively), automatic text illustration (Joshi et al., 2006), or explaining the text using the imagined scene for someone who does not understand the language the text is written in.

1.1 Previous Work Carried Out

Last year, the main focus of the project was to discover the most semantically meaningful image features which could help build models capable of creating fluent, grammatically correct and relevant to the scene linguistic descriptions. The Abstract Scenes Dataset (Zitnick and Parikh, 2013) which includes 10,020 clip art images with a set of image features already prepared was used during the course of the project to avoid using noisy features extracted with the computer vision techniques.

Multimodal neural language models (Kiros et al., 2014) were selected as the main approach for building an automatic caption generation system. The experimentation was completed to select word features and the most semantically meaningful visual features, so that the performance of multimodal neural language models is optimised. It was found that object's occurrence and object's distance to the optionally present person's hand and head features are the only features needed for semantically meaningful caption generation, whereas addition of any other features makes the models perform worse.

Two baseline methods for the task of image description generation were introduced. These included the simple language model and two multimodal neural language models trained with features extracted with computer vision techniques, namely convolutional networks. Our proposed models which were trained using a subset of carefully selected Abstract Scenes features were automatically evaluated[1] against the baselines. Finally, human based performance analysis was done manually to confirm the automatic evaluation scores achieved.

Our proposed models proved to be capable of generating fluent and relevant to the scene descriptions reliably and consistently. They have significantly outperformed both baseline models in terms of automatic evaluation and human based analysis.

Note that the image description generation project was fully completed last year, therefore we have set a new aim for the current year's project. As described in the previous sections, the current project goal is to build a system capable of generating images for the provided descriptions. This is a reversed version of the previous year's task.

Since the methodology needed for generating images is different from generating image descriptions, we do not intend to give a detailed explanation of the multimodal

[1]Evaluation metrics used include BLUE and Meteor. These methods were carefully selected to make sure that the metrics are reliable and correlated with human judgements in the complicated task of scoring generated image descriptions.

neural language models and evaluation techniques used in building image description generation model. The only matter kept from the last project and which the reader should be familiar with is the Abstract Scenes Dataset. Thus, it is described in detail in Section 2.1.

1.2 Main Contributions

Throughout the course of this project we were focusing on building coherent with the given descriptions image generation model and investigating methods for evaluating image quality. To focus on the core problem of semantic scene understanding, the Abstract Scenes Dataset (Zitnick and Parikh, 2013) was used. The use of this dataset allows to avoid problems which arise with the use of noisy automatic object detectors in real images. It also allows to simplify the task so that instead of *drawing* the imagined scene, only the clip arts and their positions in the scene have to be identified.

We propose a two step methodology for generating images. First, given descriptions are translated into visual sentences using the modified phrase-based Statistical Machine Translation (SMT) model suggested by Ortiz et al. (2015). Visual sentences serve as an intermediate language, specifying the spatial relations between objects discovered in the textual descriptions. These are then used as an input to the second step of the methodology; it first selects clip arts which are required to be positioned in the image, and finally defines the Mixed Integer Linear Program (MILP), the solution of which provides optimal variable values for each clip art, which are then used to render a complete image. High level overview of the methodology is illustrated in Figure 1.2.

Most of our work was done in order to modify and improve the phrase-based Statistical Machine Translation model, and define a complete Mixed Integer Linear Program definition.

We introduce three baseline image generation models- a strong novel approach by Zitnick et al. (2013) and two straightforward image retrieval methods (random and bag-of-words), to be compared with our proposed model. We propose a method of automatically evaluating the quality of generated images with the use of Visual Paraphrasing task and the corresponding dataset provided by Lin and Parikh (2015). Finally, human based performance analysis was manually done to confirm the automatic evaluation scores achieved.

Our proposed model significantly outperformed two image retrieval methods and was competitive with model built by Zitnick et al. (2013).

1.3 Outline

The remainder of this report is structured as follows.

Chapter 2 introduces the selected methods which are going to be used for building

Input descriptions	Jenny is sitting beside the swing set. Mike is kicking the soccer ball. The swing set is beside the apple tree.

↓ *Phrase-based SMT*

Visual sentences	hb1.21s.png hb0.2s.png.nearf.same.t.4s.png p.4s.png.near.same.p.9s.png
Identified clip arts	hb1.21s.png hb0.2s.png t.4s.png p.4s.png p.9s.png

↓ *MILP*

Figure 1.2: The high level overview of the two step image generation methodology. Input descriptions are first translated into visual sentences using phrase-based Statistical Machine Translation (SMT) model, proposed by Ortiz et al. (2015) (Section 2.2), and improved as in Section 3.1. The clip arts needed to be placed in the image are then identified (Section 3.2.1) and the Mixed Integer Linear Program (MILP) associated with input visual sentences is solved (Section 3.2.3) to produce the final image. In the example above, *hb1.21s.png* corresponds to the girl clip art, *hb0.2s.png* - boy, *t.4s.png* - ball, *p.4s.png* - swings, and *p.9s.png* - apple tree.

the image generation model. The Abstract Scenes Dataset (Zitnick and Parikh, 2013), phrase-based Statistical Machine Translation model (Ortiz et al., 2015) for translating linguistic descriptions into visual sentences, and the short overview of Mixed Integer Linear Programming are all described in this chapter.

Chapter 3 describes the full image generation model design. The modifications and improvements to phrase-based Statistical Machine Translation model (Ortiz et al., 2015) are first described. The detailed description of the second step of our methodology (image generation given visual sentences) is then given. This phase includes identifying clip arts which are to be placed in the image, computing prior belief states for clip arts and finally solving Mixed Integer Linear Program to find the necessary variable values for clip arts to be added into the image.

Chapter 4 evaluates and compares our proposed model with other baseline methods using automatic metric and manual human analysis. The application of Visual Paraphrasing task (Lin and Parikh, 2015) is proposed as a way for automatically evaluating generated images. This chapter also provides the intuition and evidence for some of the design decisions in Chapter 3. These include decisions made in both phases of our image generation model.

Chapter 5 summarises our report, contributions, and findings as well as it gives an insight into the future work for improving the performance of our image generation model.

Chapter 2

Background

This chapter presents the methodology used for analysing written descriptions, modelling clip art images, and describing the data used for building the image generation models.

Section 2.1 introduces the Abstract Scenes Dataset created by Zitnick and Parikh (2013). The dataset includes a variety of clip art images with associated descriptions, and this section explains how the Abstract Scenes Dataset can help in building models which are capable of generating coherent clip art images from given sets of simple descriptions.

This section goes on to describe a two step methodology for generating clip art images. The first method of doing so would be executed by training a phrase-based Statistical Machine Translation model (Ortiz et al., 2015) for translating given sets of linguistic descriptions into the corresponding visual sentences. The overview of this step is given in Section 2.2. The second step involves finding what objects should be included in the scene and in what positions these shall be placed. The latter is achieved when solving a Mixed Integer Linear Program which is introduced in Section 2.3.

2.1 Dataset

2.1.1 Abstract Scenes Dataset

As Zitnick and Parikh (2013) state, the aim of the Abstract Scenes Dataset is to have a set of scenes which are semantically similar. The dataset consists of 1,002 stories (1-2 sentence descriptions), each of which has 10 different scenes. Each scene has from 3 to 9 different one-sentence descriptions, capturing one specific concept in the scene.

The Abstract Scenes Dataset was created by, firstly, recruiting workers on Amazon's Mechanical Turk platform[1] to create an illustration for a children's story book by cre-

[1]The Amazon Mechanical Turk (MTurk) is a crowd sourcing Internet platform which allows recruiting individuals to perform small, paid tasks which computers are unable to do.

ating a realistic scene using a collection of clip art pieces. Workers created scenes using a graphical interface displaying a limited number of clip art objects such as toys, food, animals, and two clip art persons (children). In total, there were 80 pieces of clip arts designed, 24 of which included a person (7 different poses and 5 different facial expressions for each child - boy and a girl) and other 56 contained other objects. A simple background ($500 \times 400 px$) including grass and a blue sky was used. In order to avoid empty scenes or scenes including more than two children, workers were urged to use at least 6 clip art pieces. Each of them had to be used only once, and one boy and one girl at most had to be added to the scene. Additionally, each piece of clip art could have been flipped horizontally and scaled using three fixed sizes.

As a result, 1,002 initial scenes were created. Hereafter a new set of workers were asked to describe the scenes using one or two sentences, resulting in total of 1,002 stories. The names *Mike* and *Jenny* were provided if workers wished to name boy and girl clip art pieces in their sentences.

Subsequently, with help from the same graphical interface as before, a new set of workers were asked to generate a scene. However, this time workers were given one of 1,002 stories for which the scene of the corresponding meaning had to be created. Ten scenes were generated from each story which resulted in 10,020 scenes in total. Finally, another different set of the Mechanical Turk workers were introduced to describe one of 10,020 scenes. Every scene was described with 2 to 6 short sentences by two workers resulting in 3 to 9 captions per scene and 60,396 descriptions in total. Only these 10,020 scenes with associated 60,396 short captions were used throughout the course of this report. An example of the scene with its associated two sets of captions (by two independent workers) is shown in Figure 2.1.

2.1.2 Reasons Selecting the Dataset

There are a couple of reasons for choosing to work with the Abstract Scenes Dataset over other datasets which use realistic photos. As introduced in Chapter 1, identifying the specific parts of an image (e.g. objects, actions, and relationships) is a challenging problem, and even with a lot of effort taken to solve this problem with computer vision techniques, there is still no good solution for it. However, with use of the Abstract Scenes Dataset it is possible to avoid any image feature extraction, and hence any noise, since together with each scene the dataset includes image rendering information which explicitly specifies what objects are present in the image and in which exact locations. An example scene with its rendering information is shown in Figure 2.1. Such dataset feature also allows to completely avoid an actual image *drawing* process. All what is needed to perform in order to generate an image is finding what objects should be included in the image, and in what specific locations x and y, depths z, and facing directions d. The final image can then be easily rendered using such provided information.

As Zitnick and Parikh (2013) claim and as Heider and Simmel (1944) demonstrate through their psychological experimental research, photo realism is not necessary for the study of semantic meaning understanding, and thus any research on the Abstract

Image rendering information						
Clip art	Type id	Object id	Position x	Position y	Depth z	Flip d
s.3s.png	0	3	467	24	2	1
hb0.10s.png	2	10	145	182	0	1
hb1.19s.png	3	19	323	188	0	1
c.9s.png	5	9	161	116	0	1
c.5s.png	5	5	327	95	0	1
t.4s.png	7	4	43	173	0	0

Captions from set 1
Jenny is upset because Mike isn't sharing the soccer ball.
Mike is wearing sunglasses.
Jenny is wearing a silly hat.

Captions from set 2
Mike is kicking the soccer ball away from Jenny.
Jenny is chasing Mike to get the ball.
Jenny is wearing a silly hat.

Figure 2.1: Sample scene with the associated complete image rendering information and two sets of captions created by two independent persons. Note that $0 \leq x_i \leq 500$, $0 \leq y_i \leq 400$, $z_i \in \{0,2\}$, $d_i \in \{0,1\}$, and $x_i, y_i, z_i, d_i \in \mathbb{Z}$ for each clip art i.

Scenes Dataset should be applicable to photo-realistic images. Indeed, one could argue that the number of available photo-realistic scenes is enormous, and that they should be easily extracted from the Internet or other sources, and probably even with plausible associated descriptions which could be used for training. However, it would still be challenging to find an alternative dataset. The Abstract Scenes Dataset is relatively large[2] dataset with 60,396 fluent, short descriptions and 10,020 scenes. There is no doubt that larger datasets (Berg et al. (2010), Ordonez et al. (2011)) which include millions of images and their descriptions, and are directly scraped from the web, would include noise and unnecessary, unrelated to the image information.

Lastly, real images may contain a diverse range of scenes resulting in a sparse set of semantic concepts. On the contrary, the Abstract Scenes Dataset only use a finite set of clip arts and have a dense selection of semantically similar scenes which allows studying subtle semantic meaning nuances in the finite domain.

The reasons discussed in this section described why the Abstract Scenes Dataset is useful for the task of image generation and why it has been chosen for the purposes of this project.

2.2 Phrase-based Statistical Machine Translation (SMT) Model

In this section, a phrase-based Statistical Machine Translation (SMT) model for automatic image description generation[3] (Ortiz et al., 2015) is presented. The main steps in the process involve creating a parallel corpus by identifying and extracting visual sentences from the original linguistic descriptions in the Abstract Scenes Dataset, and then learning a phrase-based SMT model which can translate from a visual sentence into a linguistic description. More detailed explanations of these two phases are given in Sections 2.2.1 and 2.2.3. Before explaining how the phrase-based SMT model is trained, short introduction to the standard phrase-based SMT is given in Section 2.2.2.

It is argued in this section that model originally created for a task that is an opposite version of what is attempted to accomplish (generate images) can be conveniently adapted to fit our purposes. Phrase-based SMT model can be equally well trained to translate linguistic descriptions into the corresponding visual sentences by reversing the direction of the original (Ortiz et al., 2015) phrase-based SMT model.

[2] As an example, the Pascal Sentences (Rashtchian et al., 2010) and IAPR TC-12 (Grubinger et al., 2006) datasets consist of wide variety of images and high quality descriptions. However, they are limited in size (9,000 and 20,000 images respectively).

[3] The original model was used for the task of automatic image description generation and not the opposite task of generating images.

Linguistic general form	Example predicate tuple
subject relation object	Jenny is wearing a silly hat

Visual general form	Example visual tuple
clip1 f(clip1, clip2) clip2	hb1.19s.png f(hb1.19s.png, c.5s.png) c.5s.png

Figure 2.2: The parallel corpus structure of the linguistic and visual sentences for the example description *Jenny is wearing a silly hat* and the corresponding scene in Figure 2.1. Textual description in the general form is called *predicate tuple* where both, subject and object, are called *predicate objects*, and the structure of the associated visual relation is called *visual tuple*.

2.2.1 Parallel Corpus Creation

The purpose of this section is to describe a method (Ortiz et al., 2015) for creating a parallel corpus for training a phrase-based SMT model which could be used for image generation.

The methodology presented in the following sections aims to create a parallel corpus of a strict form which includes general form linguistic (*predicate tuple*) and visual (*visual tuple*) descriptions. The structure and examples of both, linguistic and visual, general forms are shown in Figure 2.2. Note that for the description to conform to the parallel corpus general form, the sentence must be in the following form - subject, relation, object. However, not all of the descriptions in the Abstract Scenes Dataset are in such structure (description *Mike is kicking the soccer ball away from Jenny* includes three predicate objects *Mike*, *soccer ball* and *Jenny*, and two relations *is kicking* and *away from*). Therefore, *consistent predicate object pairs* with the associated *consistent predicate tuples* are first extracted from linguistic descriptions which match the parallel corpus general form as described in Section 2.2.1.1.

Next, visual tuples of the form (clip1, f(clip1, clip2), clip2) are created as presented in Section 2.2.1.2. This includes mapping predicate objects (object and subject in the textual sentence) to their associated clip arts (clip1 and clip2) and encoding the spatial information between them using visual dependency representation f(clip1, clip2). Finally, the parallel corpus is obtained as described in Section 2.2.1.3.

2.2.1.1 Creating Predicate Tuples

As noted before, not all of the descriptions in the Abstract Scenes Dataset conform to the parallel corpus general form. However, most of the descriptions can be transformed into the general form and this section explains how.

First, the descriptions are parsed using a dependency parser[4] and the expressions, which function as predicate objects, are identified. Stanford dependency parser Klein and Manning (2003) was used in all experiments run for this project.

[4]Extracted dependencies provide a representation of grammatical relations between words in the sentence.

The technique for identifying predicate objects in the parsed descriptions works as follows. The subject part of the sentence is identified using the tags *nsubj* and *nsubjpass*, whereas the object part is identified using tags *dobj* and *pobj*. If either the subject or the object contains adjectives and/or noun compound modifiers, these are included into the predicate object by expanding the dependency tree down from the tagged subject/object. Note that determiners and possessives are not included into the predicate object as neither of them add valuable information when identifying clip arts[5].

As an example, the sentence *Mike is kicking the soccer ball away from Jenny* is parsed as

```
-> kicking-VBG (root)
  -> Mike-NN (nsubj)
  -> is-VBZ (aux)
  -> ball-NN (dobj)
    -> the-DT (det)
    -> soccer-NN (nn)
  -> away-RB (advmod)
  -> from-IN (prep)
    -> Jenny-NN (pobj)
```

Following the predicate object identification procedure, 3 objects are obtained: subject *Mike* and two objects - *the soccer ball* and *Jenny*. Once predicate objects are found, the sentence which realises the parallel corpus general form is extracted. The whole original sentence is returned in such case when only one or two predicate objects were found.[6] Otherwise, as happened in the previous example, *consistent predicate object pairs* are considered. A consistent pair is the one which can be used to build a natural language sentence conforming to the parallel corpus general form. Therefore, the only consistent pair in the previous example is (*Mike*, *the soccer ball*) as the resulting text *Mike is kicking the soccer ball* between these two predicate objects do indeed realise the general form sentence.[7]

By following the procedure described in this section, sentences which conform to the parallel corpus general form were identified and collected to be used for building the parallel corpus. Predicate objects are also kept for the future reference as these are going to be used for the development of the word-clip art mapping function.

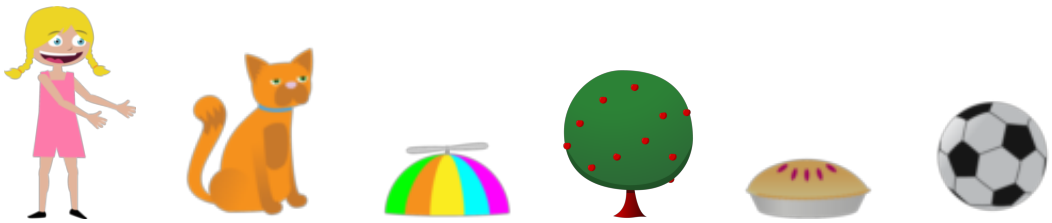
2.2.1.2 Creating Visual Tuples

There are two main steps in order to get the corresponding visual sentence in the form presented in Figure 2.2 for each linguistic general form sentence. First, predicate objects found in the textual sentence have to be mapped to their representative clip arts.

[5]Consider the phrases *the soccer ball* and *soccer ball*. Both of them refer to the same clip art regardless of the determiner *the*.

[6]There are sentences which only include a single object (*Sun is shining.*). The ones with two objects are always assumed to be in the general form.

[7]*the soccer ball away from Jenny* does not form a natural sentence, thus the object pair (*the soccer ball*, *Jenny*) is not considered as a consistent predicate object pair.



her	cat	beanie cap	apple	apple pie	ball
jenny	cats	colorful hat	apple tree	baked pie	soccer
no one	kitten	spinny hat	big apple tree	berry pie	soccer ball
pink dress	orange cat	funny hat	fruit	cake	soccer game
she	sad cat	rainbow cap	many apples	delicious pie	toe

Figure 2.3: Example predicate objects with the corresponding clip arts which have the highest mutual information.

The automatic approach used by Ortiz et al. (2015) is based on computing the mutual information between predicate objects and clip arts.[8] The mutual information between predicate objects Y and clip arts X is defined as follows:

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \, log \frac{p(x,y)}{p(x)p(y)}$$

Thus, the clip art which is used to represent the given predicate object is the one with which object's mutual information value is the highest. Figure 2.3 shows example predicate objects with the corresponding clip arts which have the highest mutual information.

Once predicate objects are mapped to the corresponding clip arts, the visual relation between subject and object in the linguistic sentence has to be encoded. The Visual Dependency Representation model outlined in Elliott and Keller (2013) is adopted to encode relations according to three geometric properties: pixel overlap, the angle between clip arts, and the distance between them. All the relations used are summarised in Table 2.1. The last three relations encode the relative object position in z space (depth), whereas all other encoding rules encode the spatial object relation in x-y space. Some sentences in the Abstract Scenes Dataset only include one predicate object. In such cases objects are encoded using singleton visual tuple which consists of the corresponding clip art only. For example, for the sentence *Sun is shining* the associated visual tuple would be *s.3s.png*.

Using the methodology described above, a visual sentence for each predicate tuple can be generated. Table 2.2 demonstrates the parallel corpus corresponding to Figure 2.1.

[8] Mutual information quantifies the amount of information gained about one random variable given the other observed random variable.

Relation	Encoding rule
X on Y	More than 50% of X overlaps with Y
X surrounds Y	X overlaps entirely with Y
X above Y	The angle between X and Y is between 225°and 315°
X below Y	The angle between X and Y is between 45°and 135°
X opposite Y	The angle between X and Y is between 315°and 45°or 135°and 225°. Euclidean distance between the objects is greater than $w \times 0.72$
X near Y	Similar to opposite but Euclidean distance between X and Y is greater than $w \times 0.36$
X close Y	Similar to opposite but Euclidean distance between X and Y is lower or equal than $w \times 0.36$
X infront Y	X is in front of Y in the z-plane
X behind Y	X is behind Y in the z-plane
X same Y	X and Y are at the same depth

Table 2.1: Visual Dependency Representation rules (Ortiz et al., 2015) for encoding the x-y (first seven) and z (last three) spatial relations between a pair of clip art objects. In cases where X is facing Y relations *opposite*, *near* and *close* are subscripted with letter f. All relations are considered with respect to the centroid of an object and the angle between these centroids (0°lies to the right and a turn around the circle is counterclockwise). All regions are mutually exclusive. Parameter *w* refers to the width of the scene.

2.2.1.3 Creating the Parallel Corpus

The process of creating a parallel corpus is a straightforward application of the methods mentioned before. Specifically, for each description in the Abstract Scenes Dataset, the first step is to extract its predicate object pairs and test if their associated predicate tuples (or some of them) realise the parallel corpus linguistic general form. In succession, the word-clip art mapping function is looked up to get the clip arts associated with the predicate objects. Then, it is checked if these clip art objects are present in the scene corresponding to the original description in the dataset.[9] If both predicate objects (subject and object) are found, a visual sentence is generated using visual dependency representation model which encodes their spatial relation. As a result, linguistic and visual sentences conforming to the parallel corpus general form are obtained.

Figure 2.4 illustrates and summarises the parallel corpus generation process. In total, 46,652 parallel sentence pairs were generated. This accounts for about 77% of all the sentences in the dataset.[10]

[9]The mapping function always outputs a clip-art object for a given text object, however there is no guarantee the returned clip art object will be present in the image as mappings are not always correct.

[10]Note that not for all of the descriptions in the Abstract Scenes Dataset the visual tuple was found due to the incorrect clip art mapping or failing to resolve parallel corpus general form.

Predicate tuple	Visual tuple
Mike isn't sharing the soccer ball	hb0.10s.png *closef same* t.4s.png
Mike is wearing sunglasses	hb0.10s.png *surrounds same* c.9s.png
Jenny is wearing a silly hat	hb1.19s.png *below same* c.5s.png
Mike is kicking the soccer ball	hb0.10s.png *closef same* t.4s.png
Jenny is chasing Mike	hb1.19s.png *closef same* hb0.10s.png
Jenny is wearing a silly hat	hb1.19s.png *below same* c.5s.png

Table 2.2: Parallel corpus corresponding to Figure‘2.1.

2.2.2 Phrase-based SMT

In this section a short introduction to the main concepts of the standard phrase-based Statistical Machine Translation is given. Ideas in this section were taken from Koehn (2010) and we encourage the reader to consult this source for having a better understanding.

The goal of the Statistical Machine Translation is to find the most likely translation $\mathbf{e}$ for the input sentence $\mathbf{f}$. More precisely:

$$\mathbf{e}_{\text{best}} = \underset{\mathbf{e}}{\operatorname{argmax}}\, p(\mathbf{e}|\mathbf{f})$$

which can be further decomposed into the translation $p(\mathbf{f}|\mathbf{e})$ and language $p(\mathbf{e})$ models using Bayes rule:

$$\mathbf{e}_{\text{best}} = \underset{\mathbf{e}}{\operatorname{argmax}}\, p(\mathbf{e}|\mathbf{f}) = \underset{\mathbf{e}}{\operatorname{argmax}}\, p(\mathbf{f}|\mathbf{e})p(\mathbf{e})$$

Translation model measures how likely is f as a source sentence for the translation e, and language model captures how likely is the translation e as a target language sentence. Most of the machine translation systems define language model as an *n-gram* language model which makes an assumption that the probability of the current word i in the sentence is only dependent on the previous $k-1$ words. Mathematically it can be specified as:

$$p(\mathbf{e}) = p(e_1, e_2, \ldots, e_n) = \prod_{i=1}^{n} p(e_i|e_{i-(k-1)}, e_{i-(k-2)}, \ldots, e_{i-1})$$

where n is the number of words in $\mathbf{e}$. Translation model, on the other hand, is more complicated and many techniques for defining it have been proposed. Phrase-based models are currently the most successful approach. They are based on splitting the input sentence into the sequence of phrases (chunks of words), mapping each phrase with exactly one output phrase and finally reordering the output phrases to produce the most natural and correct sentence possible. Formally,

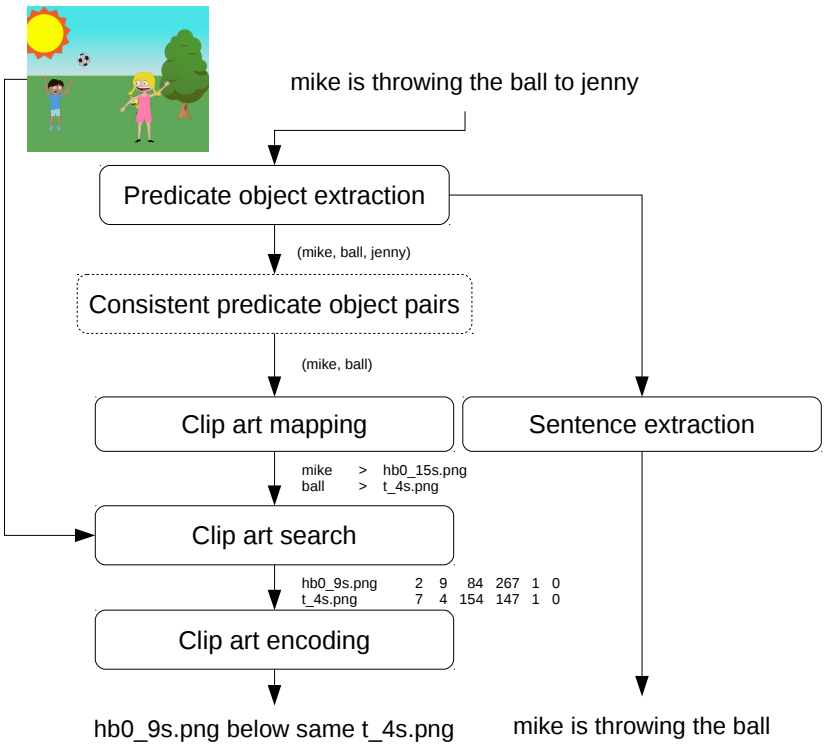


Figure 2.4: Overview of the parallel corpus generation process. Finding consistent predicate object pairs step is optional as descriptions with 2 or less predicate objects are considered as always consistent. Reprinted from (Ortiz, 2014).

$$p(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{I} \phi(f_i|e_i) d(start_i - end_{i-1} - 1)$$

where I is the number of phrases in the sentence. Each input phrase f_i is translated into an output phrase e_i using the phrase translation model ϕ. Phrases may be reordered using distance-based reordering model d which is usually implemented to penalise movement of phrases over larger distances.

Both language and phrase translation models are learned from the parallel corpus by gathering statistics about n-word target language sequences and translations of phrases. After collecting the statistic the model is ready to be used for translating input sentences into the output. In Machine Translation, this task is called *decoding* which aims to find the best scoring translation $\mathbf{e}$. It has been shown that finding the exact solution in this task is NP-complete, therefore multiple heuristics are introduced to efficiently carry out the search. As a result, there is no guarantee that the best possible translation will be found.

2.2.3 Phrase-based SMT Model Training

The task for translating textual descriptions into visual sentences is modelled as a phrase-based SMT where **f** corresponds to the input descriptions and **e** is the required output - visual sentences. The statistics required for the phrase-based SMT model are gathered from the parallel corpus created as in Section 2.2.1.3. This is done with the use of MOSES Koehn et al. (2007). MOSES is an implementation of state of the art SMT algorithms. The software is used as a configurable black box which processes parallel data to learn a machine translation model. The configuration file allows setting how the training data is prepared (tokenisation, cleaning), what are the source and target languages, their language and translation models, and what their tuning and evaluation methodology is. Such simple model is enough to serve our purposes.

The SMT model is trained by using 70% (7014 scenes) of the Abstract Scenes Dataset. 9% (901 scenes) of the data is used for tuning the phrase-based SMT model parameters, 1% (101 scene) is used for testing the phrase-based SMT model and the remaining 20% of the data is used for testing the final image generation model (2004 scenes). The resulting model is now capable of translating a novel linguistic descriptions into visual relations. MOSES software is also used for the decoding task. For instance, given a textual sentence *Mike is wearing sunglasses* the model outputs a visual sentence *hb0.10s.png surrounds same c.9s.png* which can be used for image generation in the later stage.

This section outlined the basic phrase-based SMT model for translating given linguistic descriptions into the corresponding visual sentences. We modify and improve this model as described in Section 3.1. Visual sentences acquired by phrase-based SMT model can then be used to resolve variable values for all clip arts found in the visual sentences, which effectively specifies all information needed to render a scene. This is done with help of Mixed Integer Linear Programming.

2.3 Mixed Integer Linear Programming (MILP)

Linear programming (LP) is a method for achieving the best outcome in a mathematical model which uses only linear relationships to express its requirements. Formally, linear programming aims to optimise a linear objective function subject to linear constraints which take either equality or inequality form. The solution to such problem is a set of optimised variable values which optimise linear objective. Mixed integer linear program (MILP) is a linear program in which some of the variables are constrained to be integers.

MILP can be expressed in canonical form as

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x}_i \in \mathbb{Z} \text{ for } i \in I \end{aligned}$$

where $\mathbf{c}, \mathbf{b}$ are vectors and A is a matrix, having integer values. x represents the vector of variables to be determined where I is a set of integer variables indexes. The objective function here is defined as $\mathbf{c}^{\mathrm{T}}\mathbf{x}$, constraints as $A\mathbf{x} \leq \mathbf{b}$ and bounds as $\mathbf{x} \geq \mathbf{0}$.

During the course of the project, PuLP[11] Python linear programming API was used which is an abstraction of the GLPK package[12]. The full definition of the MILP used to solve a problem of optimising the positions of objects in the scene is presented in Section 3.2.3.

[11] PuLP API can be found here: https://github.com/coin-or/pulp
[12] GLPK project site: http://www.gnu.org/software/glpk/glpk.html

Chapter 3

Abstract Scene Generation Model

In this section, the full model for generating clip art images is presented. Our model consists of the two main steps. First, the set of given linguistic descriptions is translated into visual sentences. For that, the method proposed by Ortiz et al. (2015) is used. This involves creating parallel corpus and training a phrase-based Statistical Machine Translation (SMT) model as described in Chapter 2. Since the model (Ortiz et al., 2015) was originally created for the opposite task[1] multiple parts in the parallel corpus creation procedure can be improved to suit our needs better. These improvements are outlined in Section 3.1.

Once the phrase-based SMT model is trained to translate descriptions into visual sentences, the second phase of our model is applied. At this stage objects which have to be placed into the resulting image are extracted from the visual sentences and their positions as well as depth and direction variables are optimised with respect to visual tuples provided. The full description and design principles of this methodology part are given in Section 3.2.

3.1 Step 1: From Descriptions to Visual Sentences

In this section, improvements to the phrase-based SMT model (Ortiz et al., 2015) are described. Recall that the original model was built for translating visual sentences into the linguistic descriptions. However, in order to get the model which could translate from descriptions into visual sentences is straightforward. Instead of using visual sentences as a source for phrase-based SMT model training, we use them as a target now, which ultimately leads to the required model. Fortunately, the parallel corpus is not needed to be changed, however, we make multiple improvement to it.

Recall the parallel corpus generation process as illustrated in Figure 2.4. We re-design some of these steps to accomplish a better phrase-based SMT model which would lead to a better generated images in the later stage. First, parallel corpus general form

[1]The task of generating descriptions for the given visual sentences (extracted from given clip art images).

is changed to better fit our desired model as described in Section 3.1.1. We change the way linguistic descriptions are extracted from the original sentences and slightly modify the structure of the visual sentences.

Next set of improvements correspond to the methodology of extracting predicate objects from the original linguistic sentences and finding the predicate object pairs which are consistent. These improvements are directly influencing visual sentences which are going to be created for the corresponding linguistic description. The detailed overview of the mentioned stage improvements is presented in Section 3.1.2. We leave clip art mapping and search stages unchanged from the design by Ortiz (2014).

3.1.1 Parallel Corpus General Form

Recall the parallel corpus general form as in Figure 2.2. There used to be a strict structure for both visual and textual parts of the parallel corpus. We make multiple changes in this structure to increase the performance of both phrase-based SMT and image generation models.

Visual general form, as shown in the upper table of Figure 3.1, has not been changed drastically. The main difference is the concatenation of subject, relations and object terms. Note that the aim of the visual tuples is to provide the spatial relations between two clip arts in x-y-plane and z-plane. Therefore, it is preferred that the phrase-based SMT model outputs full and consistent visual tuples which provide the required information. Random combination of objects and relations in any order would not be able to serve such need.

Contrarily, linguistic general form was modified significantly. In fact, its strict form was completely removed. We recognised that truncating the given sentence to its strict form[2] removes crucial information which could be used in generating better clip art images. Note that consistent predicate object pairs extraction procedure has not been changed up to this point. The example linguistic description (original sentence) is shown in the bottom table of Figure 3.1.

Arguably, the original textual sentence includes information which is not valuable. Possessives and determiners, for example, do not provide additional knowledge about the objects and relations in the sentence. This fact was also recognised by Zitnick et al. (2013), and hence, tuples of a strict form (subject, relation, object) were extracted from the linguistic descriptions. Only the subject, object and the main relation in the canonical (lemma) form[3] between them are considered. Therefore, adjectives which could help distinguish between clip arts are removed. An example extracted tuple is presented in the bottom table of Figure 3.1.

Due to the limitations in the tuples created by Zitnick et al. (2013), we used a Stanford dependency parser (Klein and Manning, 2003) to extract basic dependencies between

[2]Strict general form (subject, relation, object) as proposed by Ortiz et al. (2015).

[3]Canonical form (lemma) is a headword of a set of all forms of the word which have the same meaning. Words *run*, *runs*, *ran*, *running* have the same meaning with *run* as a lemma.

the words in the linguistic sentence. Figure 3.1 shows basic dependencies, dependencies in the canonical (lemma) form, and main dependencies for the example sentence. Main dependencies represent the set of dependencies which provide any valuable information[4].

Note that since linguistic general form is no longer strict it may include multiple consistent predicate object pairs. As a result, multiple visual tuples would be created for the single linguistic description and, unlike previously, would be kept together in the same visual sentence. Therefore, from this point visual sentence is no longer an equivalent for visual tuple but is rather a set of visual tuples.

We build a phrase-based SMT model using a modified parallel corpus and each proposed linguistic description structure. We used the same training, testing and tuning sets as Ortiz (2014). Finally, all built models were evaluated and compared. Evaluation was applied by calculating the precision[5], recall[6], and F_1 score[7] metrics between translated visual sentences and the corresponding gold standard sentences. Because getting the whole visual tuple correct is hard, we also assessed how well the models translate objects and relations itself. Three F_1 scores were calculated by only considering a specific part of a single visual tuple i.e. relations, object pairs, and objects itself. Consider the example sentence in Figure 3.1. To test the model's output for the input sentence *Jenny is wearing a silly hat* with respect to the gold standard visual sentence *hb1.19.png.below.same.c.5s.png* we would perform four tests. First, we would check if the whole visual tuple produced is correct. Secondly, there would be checked if the relation is the same as *below.same*. The next test would check if clip art objects pair is indeed (*hb1.19.png*,*c.5s.png*) while the final test would check if any of the clip arts produced is correct. The evaluation results are summarised in Table 3.1. The scores suggest that using main dependencies (lemmas) as a linguistic general form significantly outperforms all other alternative linguistic description structures. Therefore, main dependencies in the canonical (lemmas) form are continued to be used throughout the remainder of this report.

3.1.2 (Consistent) Predicate Object Extraction

Consider the example sentence *Mike is kicking the soccer ball away from Jenny* with the parsed dependencies as outlined in Section 2.2.1.1. Under the current model, the fact that the ball is kicked away from Jenny would be left out as predicate object *Jenny* does not constitute to any consistent predicate object pair. However, this information

[4]All of the following dependencies are removed as they do not provide any semantically valuable information: *acomp*, *aux*, *auxpass*, *cc*, *cop*, *det*, *discourse*, *expl*, *punct*, *mark*, *mwe*, *ret*, *root*, *npadvmod*, *num*, *number*, *parataxis*, *pcomp*, *possessive*, *preconj*, *quantmod*.

[5]Precision is calculated as a fraction of retrieved visual dependencies (visual sentences) which are relevant (also found in the gold standard).

[6]Recall is calculated as a fraction of the relevant visual dependencies which are successfully retrieved (translated).

[7]F_1 score considers both precision and recall as equally important factors to the evaluation and is calculated as $F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$. Note that $0 \leq F_1 \leq 1$.

Visual general form	Example visual tuple
clip1.f(clip1, clip2).clip2	hb1.19s.png.*below*.*same*.c.5s.png

Linguistic general form	Example predicate tuple
Original description	Jenny is wearing a silly hat
Tuple	Jenny wear hat
Dependencies	nsubj.wearing.Jenny aux.wearing.is det.hat.a amod.hat.silly dobj.wearing.hat
Dependencies (lemmas)	nsubj.wear.Jenny aux.wear.be det.hat.a amod.hat.silly dobj.wear.hat
Main dependencies (lemmas)	nsubj.wear.Jenny amod.hat.silly dobj.wear.hat

Figure 3.1: The modified parallel corpus structure of the visual and linguistic sentences for the example description *Jenny is wearing a silly hat*. Five alternatives for the linguistic general form are proposed. These include original sentence, tuple (Zitnick et al., 2013), and dependencies extracted from the original sentence.

is crucial, and it would be useful to include the visual relation between soccer ball and Jenny into the visual sentence representing the given linguistic description.

Next, consider another sentence *Jenny is throwing the football to Mike* which consist of the following dependencies:

```
-> throwing-VBG (root)
  -> Jenny-NN (nsubj)
  -> is-VBZ (aux)
  -> football-NN (dobj)
    -> the-DT (det)
  -> Mike-NN (vmod)
    -> to-TO (aux)
```

In this case, because *Mike* is not considered as a predicate object (since *vmod* dependencies are not considered as ones holding information about predicate objects) the information about Mike's visual relation to either *Jenny* or *football* is not examined.

To overcome these issues the predicate object extraction procedure is modified in the following way:

- In addition to the existing tags used for subject and object identification[8] tags *vmod* and *nn* are included.
- Because new tags are added into the consideration, in addition to determiners and possessives which are excluded from the predicate objects, auxiliaries are removed as well. The auxiliary *to* does not add any additional information about the object *Mike* in the predicate object *to Mike*, hence it is removed.

The search of consistent predicate object pairs is also updated and is performed depending on the number of predicate objects found in the sentence:

[8]These include *nsubj*, *nsubjpass*, *dobj* and *pobj*.

Linguistic general form	F_1 score			
	Full	Relations	Object pairs	Single objects
Original sentence	0.043	0.219	0.058	0.254
Tuple (Zitnick et al., 2013)	0.042	0.259	0.084	0.385
Dependencies	0.067	0.322	0.111	0.329
Dependencies (lemmas)	0.071	0.323	0.112	0.340
Main dependencies (lemmas)	**0.086**	**0.358**	**0.184**	**0.422**

Table 3.1: Evaluation scores for phrase-based SMT models trained using different linguistic general form in the parallel corpus. Model outputs were evaluated with respect to the gold standard visual sentences. F_1 scores were calculated considering the original outputs (Full), relations only (Relations), subject and object pairs (Object pairs), and single objects. Note that model built using main dependencies in the canonical (lemmas) form as a linguistic general form significantly outperforms all the other models in all four test.

One predicate object that object is always considered as consistent.

Two predicate objects their pair is always considered as consistent.

More than two predicate objects all *neighbouring* predicate objects reside a consistent predicate object pair. Full set of consistent predicate object pairs for the examples above are (*Mike*, *soccer ball*) and (*soccer ball*, *Jenny*) for the first one, and (*Jenny*, *football*) and (*football*, *Mike*) for the second one.

We call this model an updated version of the best phrase-based SMT model, found in the previous section. Two more models are considered with an additional modification to the current model. The modifications for each additional model are described below:

Updated + combinations in this model, all combinations of predicate objects are considered as consistent pairs instead of only considering *neighbouring* predicate objects.

Updated + singleton we notice that some sentences have multiple predicate objects from which only a single one corresponds to the existing clip art. Consider an example *There is a plane in the sky.* Both *plane* and *sky* are predicate objects, and form a consistent pair. However *sky* does not have a corresponding clip art in the dataset, thus it is likely that no visual tuple will be created for such sentence and the knowledge about *plane* will be lost. This model ensures that each predicate object found in the sentence is included in a singleton visual tuple if it was not a part of any consistent predicate object pair.

Visual sentences extracted for the example descriptions using the final model from Section 3.1.1 and three models described in this section are presented in Figure 3.2. Models introduced in this section were trained in the same way as the previous models and the results are summarised in Table 3.2. As can be seen from the results, none

	F_1 score			
Linguistic general form	Full	Relations	Object pairs	Single objects
Main dependencies (lemmas)	0.086	**0.358**	0.184	**0.422**
Updated	0.083	0.344	0.184	0.392
Updated + combinations	0.076	0.328	0.159	0.368
Updated + singleton	**0.120**	0.287	0.180	0.342

Table 3.2: Evaluation scores for phrase-based SMT models trained using the final model from Section 3.1.1 and three model described in Section 3.1.2. Note that it is not clear which model is the best.

of the models is a clear winner. Since producing a good translation model is not the main objective of our task, we build a complete image generation model with each of these phrase-based SMT models and only then decide which one helps the next stage to produce the best images. This part of the evaluation is presented in Chapter 4.

3.2 Step 2: From Visual Sentences to Clip Art Images

Once the phrase-based SMT model for translating linguistic descriptions into visual sentences is designed and trained, the second step in the pipeline is to generate clip art images.

Given a set of visual sentences, objects which are going to be placed into the clip art image are determined first. The outline of this procedure is given in Section 3.2.1.

Next, we define the Mixed Integer Linear Program (MILP) to optimise the position, depth, and facing direction variables for the objects identified before. The optimisation is performed with respect to the visual tuples which are provided as well as prior belief positions for each clip art and relative positions for each clip art pair. The last two are calculated from the training set's clip art images and are described in detail in Section 3.2.2. Finally, the MILP definition is given in Section 3.2.3.

3.2.1 Selecting Objects

Given the set of visual sentences, the first task is to find the set of objects which should be represented in the corresponding clip art image. Since visual sentences already include the exact clip arts needed, all unique objects and their clip arts can be easily extracted.

Note that there might be multiple different clip arts in the given set of visual sentences corresponding to the same boy or girl but in different pose or with a different facial expression. However, only one boy or girl is allowed to appear in the image, thus the decision has to be made which particular clip art representing boy or girl should be selected.

Mike is kicking the soccer ball away from Jenny	
Model	**Visual sentence**
Main dependencies (lemmas)	hb0.10s.png.closef.same.t.4s.png
Updated	hb0.10s.png.closef.same.t.4s.png t.4s.png.nearf.same.hb1.19s.png
Updated + combinations	hb0.10s.png.closef.same.t.4s.png hb0.10s.png.close.same.hb1.19s.png t.4s.png.nearf.same.hb1.19s.png
Updated + singleton	hb0.10s.png.closef.same.t.4s.png t.4s.png.nearf.same.hb1.19s.png

Jenny is throwing the football to Mike	
Model	**Visual sentence**
Main dependencies (lemmas)	hb1.24s.png.close.behind.t.6s.png
Updated	hb1.24s.png.close.behind.t.6s.png t.6s.png.near.infront.hb0.3s.png
Updated + combinations	hb1.24s.png.close.behind.t.6s.png hb1.24s.png.nearf.same.hb0.3s.png t.6s.png.near.infront.hb0.3s.png
Updated + singleton	hb1.24s.png.close.behind.t.6s.png t.6s.png.near.infront.hb0.3s.png

There is a plane in the sky	
Model	**Visual sentence**
Main dependencies (lemmas)	
Updated	
Updated + combinations	
Updated + singleton	s.7s.png

Figure 3.2: Visual sentences extracted for the example descriptions using the final model from Section 3.1.1 (*Main dependencies (lemmas)*) and three models described in Section 3.1.2. Note the difference between all these models. *Updated* model includes visual tuples for more predicate object pairs. *Updated + combinations* additionally includes all combinations of predicate objects whereas *Updated + singleton* includes singleton visual tuples for predicate objects that did not compose a consistent pair with any other predicate object.

We use Algorithm 1 to select the clip arts for both children. Note that children clip art is a combination of two properties: pose (7 different variants) and facial expression (5 different variants). In short, the most frequently appearing pose and facial expression are selected by combining the counts of the most similar poses and facial expressions.

For any other (neither girl nor boy) found object if it appeared more than once we assume it is the same object. As a result, the set of N unique objects is identified. This set is then passed to the final stage which computes the optimal position, depth, and orientation in the image for each of these objects.

3.2.2 Computing Prior Belief States

Before defining the MILP some work needs to be done beforehand. The main goal of the MILP is to find the given N objects placement in the image such that all constraints in the visual sentences are satisfied. However, just satisfying these constraints is not enough. One of the obvious problems is the fact that relations in visual tuples do not specify absolute clip art positions. Section 4.3.2 explains and Figure 4.3 illustrates other problems arising when only considering constraints defined by provided visual sentences in more detail. Consequently, additionally to satisfying the main[9] constraints, we propose to minimize the distance between the clip art i to be placed in the image and its average x, y positions in the training data for each possible depth $z_i \in \{-1,0,1\}$[10] so that clip arts are prioritised to be placed in their natural positions (for example, sun should appear up in the sky, trees should not be hanging in the air but standing on the ground).

Therefore, we compute prior belief positions for each clip art i by simply calculating its average x and y positions in the training set for each $z_i \in \{-1,0,1\}$ and defining the corresponding parameters X_i^z, Y_i^z. Figure 3.3 illustrates average positions for multiple clip arts in depth $z = 0$. The boundaries of the area 1, 2 and 3 standard deviations around the average position is also shown in the figure. These are drawn with respect to the central point of the clip art. Note that the area up to the 3$^{\text{rd}}$ standard deviation already covers 99.6% of the clip art occurrences in the training set. As can be seen from the figure, *sun* clip art appears on the very top of the image and *apple tree* is always based on the ground. These findings show that prior belief clip art positions are an important factor when deciding where the objects should be placed in the image.

Consider now a situation where two clip arts constantly appear at the similar relative position in the image. The example situation could be a girl and a hat on her head or boy and a hamburger in his hand. Note that visual tuples are not capable of capturing such information since the area defined by the relations in Table 2.1 is usually too wide. Therefore, it is naive to expect the MILP to place hat on the girl's head even if the relation *above.same* between these two objects is present.

To overcome this issue, the prior belief relative positions are defined by finding clip

[9] Main constraints are going to be called the constraints defined by visual sentences.

[10] Note that originally $z_i \in \{0,1,2\}$, however to simplify the linear equations in the later stages each z_i is shifted by -1 and thus, $z_i \in \{-1,0,1\}$. z_i values are normalized once MILP is solved.

Algorithm 1 Children clip art selection from the set of visual sentences $\mathcal{V}$ algorithm

procedure CHILDREN–CLIP–ART–SELECTION($\mathcal{V}$)
    CLIP–ARTS = {}
    for each child i **do**
        SIT(i) = the number of sitting i clip arts in $\mathcal{V}$
        KNEEL(i) = the number of kneeling i clip arts in $\mathcal{V}$
        KICK(i) = the number of kicking i clip arts in $\mathcal{V}$
        RUN(i) = the number of running i clip arts in $\mathcal{V}$
        WAVE(i) = the number of waving i clip arts in $\mathcal{V}$
        HANDS–IF(i) = the number of i clip arts with hands in front in $\mathcal{V}$
        HANDS–A(i) = the number of i clip arts with hands above in $\mathcal{V}$
        STILL(i) = WAVE(i) + HANDS–IF(i) + HANDS–A(i)
        STAND(i) = KICK(i) + RUN(i) + STILL(i)
        if STAND(i) + SIT(i) + KNEEL(i) == 0 **then**
            CLIP–ARTS(i) = NONE
            continue
        end if
        if STAND(i) $\leq$ SIT(i) + KNEEL(i) **then**
            POSE(i) = $argmax\{$ SIT(i), KNEEL(i) $\}$
        else
            POSE(i) = $argmax\{$ RUN(i), KICK(i), STILL(i) $\}$
            if POSE(i) == STILL **then**
                POSE(i) = $argmax\{$ WAVE(i), HANDS–IF(i), HANDS–A(i) $\}$
            end if
        end if
        ANGRY(i) = the number of angry i clip arts in $\mathcal{V}$
        UNHAPPY(i) = the number of unhappy i clip arts in $\mathcal{V}$
        SURPRISED(i) = the number of surprised i clip arts in $\mathcal{V}$
        LAUGHING(i) = the number of laughing i clip arts in $\mathcal{V}$
        SMILING(i) = the number of smiling i clip arts in $\mathcal{V}$
        FACE(i) = $argmax\{$ ANGRY(i), UNHAPPY(i), SURPRISED(i), LAUGHING(i), SMILING(i) $\}$
        CLIP–ARTS(i) = $\{$ POSE(i), FACE(i) $\}$
    end for
    return CLIP–ARTS
end procedure

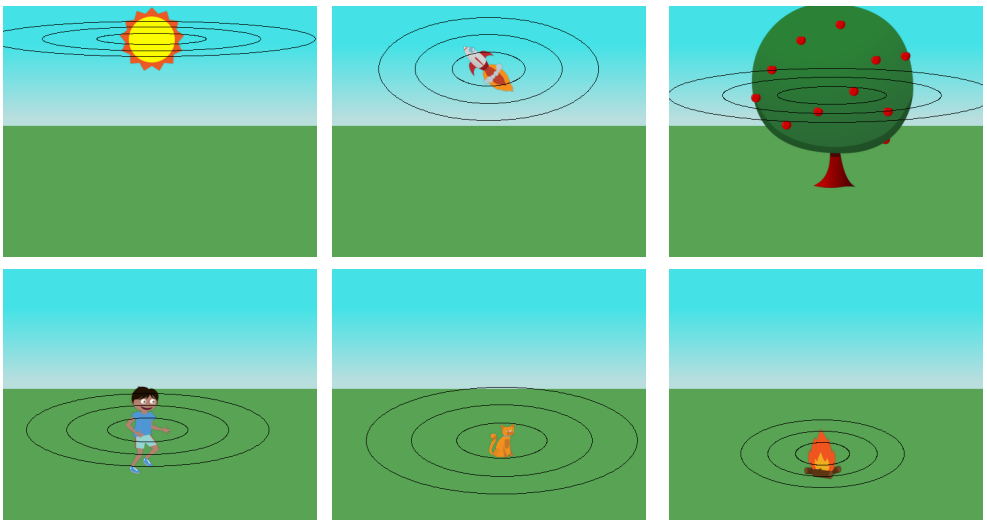


Figure 3.3: Prior belief positions for the example clip arts in depth $z = 1$. The boundaries of the area 1, 2, and 3 standard deviations around the average position illustrates where the clip art appears 99.6% of the time. Boundaries are drawn considering the centre point of the clip art.

art pairs that often appear at the similar relative locations in the training set's images and finding the average distance between them[11] in both x and y positions. Distance can be either positive or negative so that it is easy to determine which of the clip arts should be above another and which should be to the left of the other. Only the object pairs which are within 80 pixels from each other are included into the calculations. In this way, cases where two objects are unrelated are avoided[12]. Again, the same clip art appearing at the different scales z is considered separately. Hence, there are $3^2 = 9$ different possible prior belief relative positions for each two clip arts i and j and each coordinate x and y. Luckily, we can reduce this number to only 5 different parameters per coordinate:

$R^X_{-1,-1}$ and $R^Y_{-1,-1}$ — when both, clip art i and clip art j are in the same depth the distance between their x and y coordinates contribute to these parameters. Distance is scaled as for the objects in depth -1 so that it does not matter in what depth both clip arts are.

$R^X_{-1,0}$ and $R^Y_{-1,0}$ — similar to the previous parameters. These are influenced by the clip art pairs i, j for which clip art i is in depth -1 and j is in depth 0 or i is in depth 0 and j is in depth 1.

$R^X_{0,-1}$ and $R^Y_{0,-1}$ — similar to the previous. These are influenced by the clip art pairs i, j for which clip art i is in depth 0 and j is in depth -1 or i is in depth 1 and j is in depth 0.

$R^X_{-1,1}$ and $R^Y_{-1,1}$ — these are only influenced by the clip art pairs i, j for which clip art i is in depth -1 and j is in depth 1.

[11] Distance between two clip arts is calculated with respect to the clip art centre positions.

[12] For example, it would not make sense to include the relative position between the hat and the boy into average relative position calculation when the hat is on the girl's head.

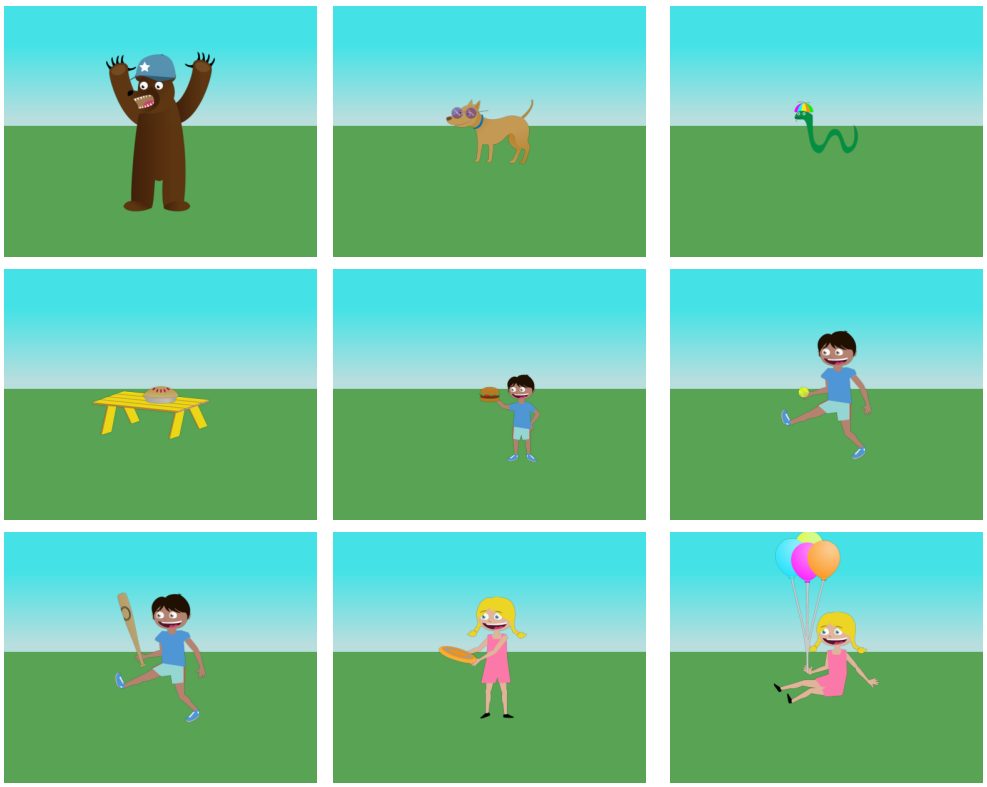


Figure 3.4: Prior belief relative positions for the example clip art pairs as learned from the training dataset by simply calculating the average distance between clip arts.

$R^X_{1,-1}$ and $R^Y_{1,-1}$ these are only influenced by the clip art pairs i, j for which clip art i is in depth 1 and j is in depth -1.

To avoid unreliable parameters, prior belief relative position parameters, for which standard deviation is more than 4 pixels in either x or y direction, are disregarded. The prior belief relative positions for the example clip art pairs are shown in Figure 3.4. The examples clearly indicate the capability of the training dataset to reliably predict the co-position of two clip arts.

3.2.3 Mixed Integer Linear Program Definition

Given the set of N clip arts which need to appear in the image, the set of visual sentences $\mathcal{V}$ which need to be satisfied, and prior belief states learned from the training data the Mixed Integer Linear Program (MILP) is defined and solved. This section provides the definition of the program used to find the optimal positions x_i and y_i, depth z_i, and direction d_i variables for each given clip art $i \in N$. The obtained information can then effectively be used to render clip art image.

We define the full set of variables as $\Theta = \{\Theta_i \text{ for } i \in N\}$ where $\Theta_i = \{x_i, y_i, z_i, d_i\}$. The overall form of the MILP with the objective of satisfying all constraints in the given visual sentences and placing clip arts in their prior belief (relative) positions is then defined follows:

$$\begin{aligned}
&\underset{\Theta}{\text{minimize}} \quad \sum_{i \in N} f(\Theta_i) + \sum_{(i,r_{x,y},r_z,j) \in \mathcal{K}} \alpha g_{r_{x,y}}(\Theta_i,\Theta_j) + \alpha g_{r_z}(\Theta_i,\Theta_j) + \beta h_{r_z}(\Theta_i,\Theta_j) \\
&\text{subject to} \quad 0 \leq x_i \leq 500,\ 0 \leq y_i \leq 400,\ -1 \leq z_i \leq 1,\ 0 \leq d_i \leq 1,\ z_i, d_i \in Z,\ i \in N
\end{aligned}$$

where N is a set of clip arts given, $\mathcal{K}$ is a set of all visual tuples $(i, r_{x,y}, r_z, j) \in \mathcal{K}$ in all visual sentences provided. Note that i and j identify both clip arts in the visual tuple while $r_{x,y}$ and r_z identify both relations (in x-y and z planes). Functions $f(\Theta_i)$, $g_r(\Theta_i,\Theta_j)$, and $h_{r_z}(\Theta_i,\Theta_j)$ correspond to the prior belief position, main, and prior belief relative position constraints. Note that function g_r is different for each relation r, since each relation defines a different set of constraints. As presented in Section 3.2.2, prior belief relative positions vary depending on the depth relation between clip arts i and j. Therefore, function h_{r_z} is defined separately for each z-plane relation. The bounds for each variable are additionally set. Both x_i and y_i are forced to lie between the image boundaries, whereas z_i and d_i are forced to take on one of the three depth values and one of the two facing values respectively. Both z_i and d_i are defined as integer variables. We do not model x_i and y_i as integer variables for computational efficiency reasons. These are rounded to the integer values once at optimum. Both α and β are used to define the importance of main and prior belief relative position constraints.

We now define the functions introduced. As stated, $f(\Theta_i)$ corresponds to the prior belief position constraints. Constraints defined by this function are used to minimize the distance between each clip art i positions x_i and y_i, and its most likely to appear positions X_i^z and Y_i^z when the clip art is at depth z_i. To minimise the distance, the absolute values of the distances between current positions and the most likely positions $|x_i - X_i^z|$ and $|y_i - Y_i^z|$ are considered. The complete function $f(\Theta_i)$ is defined as

$$\begin{aligned}
f(\Theta_i) = \quad & 0.002 \times |x_i + 0.5(X_i^{-1} - X_i^1)z_i - 0.5(X_i^{-1} + X_i^1 - 2X_i^0)|z_i| - X_i^0| + \\
& 0.0025 \times |y_i + 0.5(Y_i^{-1} - Y_i^1)z_i - 0.5(Y_i^{-1} + Y_i^1 - 2Y_i^0)|z_i| - Y_i^0|
\end{aligned}$$

where two constants 0.002 and 0.0025 normalise the errors along both directions x_i and y_i, so that both have error of maximum 1[13] and are equally important.

Next, $g_r(\Theta_i,\Theta_j)$ functions for each relation r that can appear in visual tuples are defined. The constraints are created with the reference to the rules used to represent visual relations between clip arts. These were outlined in Table 2.1. Constraints equivalent to the rules in this table are first presented:

- If $r =$ *infront* (object i is in front of object j in z-plane):

$$z_i < z_j$$

- If $r =$ *behind* (object i is behind object j in z-plane):

$$z_i > z_j$$

[13] The error along x_i can be 500 at maximum whereas error along y_i can be 400 at maximum. Normalisation pushes both of these maximum errors to 1.

- If $r = same$ (object i and object j are in the same depth):

$$z_i = z_j$$

- If $r = surrounds$ (the entirety of object j overlaps with object i):

$$|x_j - x_i| \leq RX_i \times s_i - RX_j \times s_j$$

$$|y_j - y_i| \leq RY_i \times s_i - RY_j \times s_j$$

$$s_i = -0.255 \times z_i + 0.045 \times |z_i| + 0.7$$

$$s_j = -0.255 \times z_j + 0.045 \times |z_j| + 0.7$$

where RX_i and RY_i correspond to the half of the width and height of object i clip art and s_i correspond to the scalier of the object i clip art size that denotes how much smaller object i is if it is placed at depth z_i[14].

- If $r = on$ (more than 50% of the pixels of object i overlap with object j):

$$RX_j \times s_j - RX_i \times s_i < |x_j - x_i| \leq RX_j \times s_j$$

$$RY_j \times s_j - RY_i \times s_i < |y_j - y_i| \leq RY_j \times s_j$$

which is an approximation specifying that the distances between two objects x and y positions must be smaller than object j half width and half height, and it might happen that less than 50% of the i object overlap with object j. However, to ensure that it is always more than 50% covered a non linear constraint would be required that couples both x and y variables. We realise that this problem is not crucial and it does not have any effect on the quality of generated images. s_i and s_j are calculated as before.

- If $r = above$ (object i is above object j and the angle between these two objects is between 225° and 315°):

$$y_j \geq x_j + y_i - x_i$$

$$y_j \geq -x_j + y_i + x_i$$

$$y_j > y_i + RY_i \times s_i$$

$$y_j > y_i + RY_j \times s_j$$

where s_i and s_j are calculated as before.

- If $r = below$ (object i is below object j and the angle between these two objects is between 45° and 135°):

$$y_j \leq x_j + y_i - x_i$$

$$y_j \leq -x_j + y_i + x_i$$

$$y_j < y_i - RY_i \times s_i$$

$$y_j < y_i - RY_j \times s_j$$

where s_i and s_j are calculated as before.

[14]Note that $z_i \in \{-1, 0, 1\}$ and the actual corresponding scalier $s_i \in \{1, 0.7, 0.49\}$.

- If $r = opposite$ (distance between object i and object j is bigger than 360px and the angle between these two objects is between 315° and 45° or between 135° and 225°):

$$y_j > y_i - |x_j - x_i|$$
$$y_j < y_i + |x_j - x_i|$$
$$|x_j - x_i| > 360$$

- If $r = near$ (distance between object i and object j is smaller or equal than 360px but bigger than 180px and the angle between these two objects is between 315° and 45° or between 135° and 225°):

$$y_j > y_i - |x_j - x_i|$$
$$y_j < y_i + |x_j - x_i|$$
$$180 < |x_j - x_i| \leq 360$$

- If $r = close$ (distance between object i and object j is smaller or equal than 180px and the angle between these two objects is between 315° and 45° or between 135° and 225°):

$$y_j > y_i - |x_j - x_i|$$
$$y_j < y_i + |x_j - x_i|$$
$$RX_i \times s_i \leq |x_j - x_i| \leq 180$$
$$RX_j \times s_j \leq |x_j - x_i|$$

  where s_i and s_j are calculated as before.

- If $r \in \{oppositef, nearf, closef\}$ (object i is facing object j) then we additionally to the corresponding relation above include:

$$d_i - 1 \leq 0.002 \times (x_i - x_j) \leq d_i$$

  where d_i is object i direction variable $d_i \in \{0,1\}$ and $x_i - x_j$ is normalized to lie between $\{-1,1\}$.

Note that if all of these constraints which are acquired from the visual sentences were included into MILP as actual constraints, the likelihood of infeasibility in this problem would increase. There is no doubt that visual sentences translated from linguistic descriptions by phrase-based SMT model are not perfect. Moreover, there might be contradictory linguistic descriptions provided in the first place. Either way, there is a chance that contradictory visual tuples are given as input to MILP and if these dependencies were treated as hard constrains no solution would be found for such cases.

Consequently, we make a decision to relax all main constraints, i.e move them into the objective by defining function $g_r(\Theta_i, \Theta_j)$ for each visual relation r as

$$g_r(\Theta_i, \Theta_j) = \sum_{c \in C_r} g_c(\Theta_i, \Theta_j)$$

where $\mathcal{C}_r$ is the set of all constraints for relation r as defined before and $g_c(\Theta_i, \Theta_j)$ is the relaxed constraint c. The constraint c relaxation is applied using a simple rule:

$$g_c(\Theta_i,\Theta_j) = \begin{cases} |w(\Theta_i,\Theta_j)| & \text{if c is an equality constraint } c \iff w(\Theta_i,\Theta_j) = 0 \\ |w(\Theta_i,\Theta_j) + b_c^k| & \text{if c is an inequality constraint } c \iff w(\Theta_i,\Theta_j) \leq 0 \\ |w(\Theta_i,\Theta_j) - b_c^k| & \text{if c is an inequality constraint } c \iff w(\Theta_i,\Theta_j) \geq 0 \end{cases}$$

where $b_c^k \geq 0$ is the new non negative variable and k is the id of the visual tuple[15].

As an example, consider relations *same* and *above*. Their corresponding functions $g_{\text{same}}(\Theta_i, \Theta_j)$ and $g_{\text{above}}(\Theta_i, \Theta_j)$ are defined as

$$g_{\text{same}}(\Theta_i,\Theta_j) = g_{\text{same}}(z_i,z_j) = |z_i - z_j|$$

$$\begin{aligned} g_{\text{above}}(\Theta_i,\Theta_j) = g_{\text{above}}(x_i,y_i,x_j,y_j) = \; & |y_j - x_j - y_i + x_i - b_1^k| + |y_j + x_j - y_i - x_i - b_2^k| + \\ & |y_j - y_i - RY_i \times s_i - b_3^k| + |y_j - y_i - RY_j \times s_j - b_4^k| \end{aligned}$$

$$b_1^k \geq 0,\ b_2^k \geq 0,\ b_3^k > 0,\ b_4^k > 0$$

where both s_i and s_j are defined as before and are *not* relaxed, since they define the strict relation between clip art depths and their actual scales. Other relations are relaxed in a similar way.

Finally, $h_{r_z}(\Theta_i, \Theta_j)$ function for each z-plane relation is introduced. Recall that the objective of h_{r_z} is to place two clip arts to their prior belief relative position. Note that this is done only for clip art pairs for which there is a strong correlation between their positions i.e. there exists at least one prior belief relative position parameter (as defined in Section 3.2.2) corresponding to the z-relation between the clip arts.

Each $h_{r_z}(\Theta_i, \Theta_j)$ function is a relaxation of the constraints which define the distances between two clip art coordinates $x_i - x_j$ and $y_i - y_j$ to be equal to the prior belief relative positions (distances) $\pm R^X_{z_i,z_j}$ and $R^Y_{z_i,z_j}$[16] respectively, and constraint defining that both clip arts i and j should face the same direction, precisely $d_i = d_j$[17].

In full, prior belief relative positions constraints for $h_{\text{same}}(\Theta_i, \Theta_j)$ are

$$x_i - x_j = -R^X_{z_i,z_j} \times s_i + 2 \times d_i \times s_i \times R^X_{z_i,z_j} \approx -0.73 \times R^X_{z_i,z_j} + 1.46 \times R^X_{z_i,z_j} \times d_i$$

$$y_i - y_j = R^Y_{z_i,z_j} \times s_i$$

[15] New variables are marked with both constraint identifier and the visual tuple ID to make sure that they are unique.

[16] Note that prior belief relative x positions can be either negative or positive depending on the facing variable d_i. Prior belief relative x position parameters are calculated as the clip arts are facing left $d_i = 1$. Therefore, if clip arts are facing other direction when the prior belief relative x position parameter should be swapped as well.

[17] The assumption is made that if two clip arts are correlated in their corresponding positions then they should face the same direction. This is mainly for complexity reasons and it does not affect any critical situations.

$$d_i = d_j$$

Recall that these constraints are added when depth variables of both clip arts i and j are the same $z_i = z_j$ (as it is specified by the relation *same*). Therefore, these constrains were defined to be valid only for the set ups where $z_i = z_j$. Note that the first constraint is not linear as $d_i \times s_i$ is present. We use linear regression to approximate this expression as linear function and the resulting approximate constraint is shown above. Note that this approximation ignores clip art scalier, thus slight error in the resulting prior belief relative position will be present for clip art depth $z_i = -1$ and $z_i = 1$. Similarly, the constraints for the $h_{\text{infront}}(\Theta_i, \Theta_j)$ can be defined as:

$$x_i - x_j = d_i \times (R^X_{-1,1} + R^X_{-1,0} \times s_i) - |d_i - z_j|(R^X_{-1,1} - R^X_{-1,0} \times s_i) - R^X_{-1,0} \times s_i$$
$$\approx d_i \times (R^X_{-1,1} + 0.85 \times R^X_{-1,0}) + |d_i - z_j|(0.85 \times R^X_{-1,0} - \times R^X_{-1,1}) - 0.85 \times R^X_{-1,0}$$
$$y_i - y_j = z_i(R^Y_{0,1} \times 0.7 - R^Y_{0,2}) + z_j(R^Y_{0,2} - R^Y_{0,1}) + 1.7 \times R^Y_{0,1} - R^Y_{0,2}$$
$$d_i = d_j$$

Note that because these constrains are defined for all the following depth (z_i, z_j) value combinations, $(-1,0)$, $(0,1)$, $(-1,1)$, they are even more complicated. Due to the non linearity in $d_i \times s_i$ and $|d_i - z_j| \times s_i$ terms, the constraint is approximated in the similar way as before.

All constraints for prior belief relative positions are relaxed into $h_{r_z}(\Theta_i, \Theta_j)$ function identically as it was done for main constraints. Note that $h_{\text{behind}}(\Theta_i, \Theta_j) = h_{\text{infront}}(\Theta_j, \Theta_i)$.

Note that prior belief relative position parameters might be missing if there was not enough support in defining them in the training data. In those cases, the constraints above are ignored if all required parameters are missing. If at least one of them is present then the others are set to a very large constant, so that the depth values z_i, z_j defined by the existing parameters $R^X_{z_i,z_j}$ or $R^Y_{z_i,z_j}$ are hugely preferred. We also additionally define a prior belief relative position parameter back-off function which, in case some parameter for clip art pair is missing, checks if such parameter exists for other similar clip art pair and use that as an alternative parameter option. For example, we would expect the prior belief parameters to be similar for football and basketball balls or for pink glasses and sunglasses. Similar clip art pairs are defined manually.

Not all of the constraints are fully linear yet. Many of them include absolute values, however they can be removed and the basic concepts for this procedure are provided in Appendix A. Absolute values are removed using this methodology before relaxing the constraints. Finally, the constraint importance parameters α and β were not broadly investigated. We used $\alpha = 1$, and $\beta = 0.01$ (for y-position prior belief relative position constraints) and $\beta = 0.008$ (for x-position prior belief relative position constraints). We prioritized main constraints provided by visual sentences as the most important. Prior belief relative position constraints were prioritized over prior belief position constraints. Intuitively, if two clip arts are needed to appear at some specific co-position, then most likely position for both of these clip arts is less important. Experimentation with constraint importance is part of the future work.

Chapter 4

Experiments and Results

The variability of coherent and accurate images is very high. Humans tend to visually imagine the same description differently, since there are many different image set ups which represent the meaning of the linguistic sentence equally well. Recall the situation where ten different people are asked to generate a scene for a single description as shown in Figure 1.1. It is easy to note that all of the images describe the same situation, however they are all different in a sense that different additional objects are used to make an image more elaborated and clip arts are all placed in different positions in the scenes.

The level of variability in the scenes make the evaluation task challenging. There is no automatic metric available that would be capable of measuring how good the image is with respect to the given linguistic descriptions. However, we adapt the visual paraphrasing dataset created and implemented by Lin and Parikh (2015) to achieve a reliable metric for evaluating such task. Visual Paraphrasing dataset is presented in detail in Section 4.1. The performance of our model as evaluated using Visual Paraphrasing task is presented in Section 4.2 which also compares the model with other baseline approaches.

To help interpret automatic evaluation scores the qualitative tests are performed in Section 4.3. These tests are done manually by exploring the nature of the images generated by different models.

4.1 Visual Paraphrasing as an Evaluation Metric

The Visual Paraphrasing (VP) task is to decide whether two linguistic descriptions are describing the same situation or two different ones. The expected answer to such task is a True\False decision. Lin and Parikh (2015) argue that if it were possible to first *imagine* the scene behind the text and then, in addition to reasoning about text using textual common sense, reason about the generated scenes using visual common sense, then it should help to increase the performance of the current VP models.

Therefore, Lin and Parikh (2015) experiment with this idea by creating the VP dataset

using the Abstract Scenes Dataset. Recall that each scene in the Abstract Scenes Dataset is described by two people using 2 to 6 short sentences each. Lin and Parikh (2015) use the image generation method *Full-CRF* proposed by Zitnick et al. (2013) to *imagine* a scene for each of the two set of descriptions. As a result, two scenes are generated for each of the 10,020 ground truth scenes with respect to the two sets of corresponding descriptions. These scenes are now added into the VP dataset as a positive pairs (for each pair, since they correspond to the same ground scene the VP model would be expected to realise that both scenes represent the same situation). Additionally, $2 \times 10{,}020$ pairs are randomly sampled as negatives (two scenes which correspond to a different situation). Thus, there are 30,060 questions added in to the VP dataset in total.

The model for solving VP task is trained on the 24,000 of these questions and the remaining 6,060 are left for testing. 2,020 of these are positive pairs. Briefly, the model en-corporates joint textual and visual features to classify two given sets of descriptions as describing the same situation or not. Weighted sum of features contribute to the scoring function $\Omega(I_i, S_i)$ which measures the plausibility of the given textual description S_i and the corresponding generated scene I_i as

$$\Omega(I_i, S_i) = \Phi(S_i) + \Phi(I_i) + \Psi(I_i, S_i)$$

where term $\Phi(S_i)$ captures textual common sense, $\Phi(I_i)$ captures visual common sense and $\Psi(I_i, S_i)$ captures how consistent the imagined scene is to the corresponding linguistic description. Lin and Parikh (2015) give a detailed explanation of each of these potentials. Features for learning a binary Support Vector Machine (SVM) classifier are acquired by pairing the generated scenes with the other description i.e. computing $\Omega(I_{i1}, S_{i2})$ and $\Omega(I_{i2}, S_{i1})$[1]. Trained SVM model can be used to determine whether the input pair of descriptions with the associated *imagined* scenes are describing the same situation or not. Lin and Parikh (2015) confirm that VP task performance increased from 94.15% (for model that only consider textual descriptions) up to 95.55% (for model that additionally *imagines* a scene and uses visual features).

The VP model by Lin and Parikh (2015) can now be applied as an evaluation metric for our task. Specifically, we train the VP model in the exact same way as described above except the scoring function has only the visual common sense factor remaining in it:

$$\Omega(I_i, S_i) = \Phi(I_i)$$

We re-define the model which solves the original VP task[2] into the model which solves the task of deciding whether two images are similar[3] *on their own*. Neither textual common sense nor image-text consistency is needed as we do not want the decision

[1]Generated scene I_{i1} for description S_{i1} is paired with description S_{i2} that corresponds to the other set of descriptions for the same ground truth scene. Similarly, scene I_{i2} is paired with description S_{i1}.

[2]Recall that original VP task goal is to decide whether two linguistic descriptions are describing the same situation.

[3]i.e.describe the same situation in the scene.

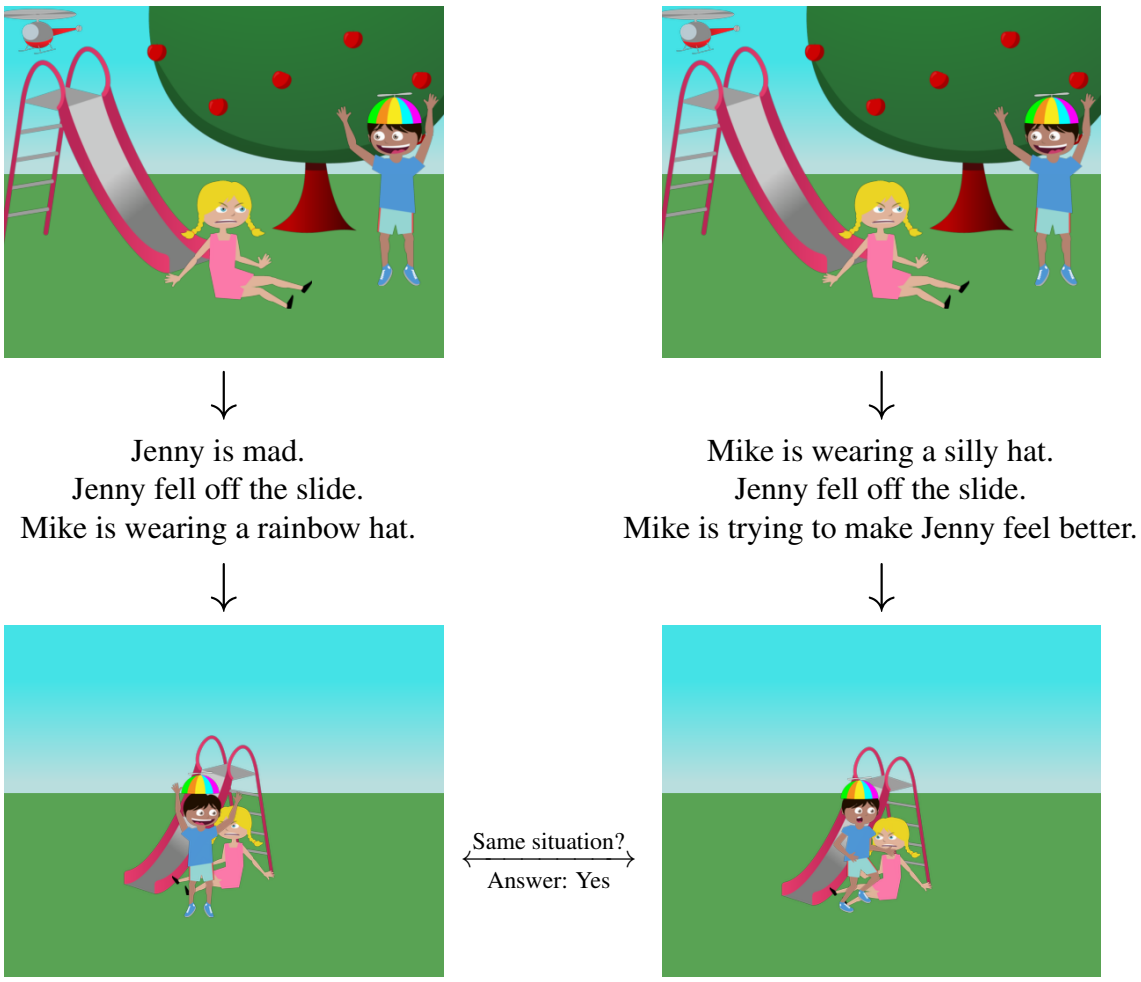


Figure 4.1: Visual Paraphrasing (VP) based testing procedure of the two scenes generated for a two sets of descriptions. Two sets of descriptions for a ground truth scene (scenes on top of the figure) are kept separately and novel scene is generated for each set of sentences. Trained VP model is applied to find out whether two scenes do indeed represent the same situation.

to be influenced by the given descriptions. We use this trained VP model to evaluate scenes generated using our method and other baselines by checking whether two generated scenes do indeed represent the same situation or not. Testing is done similarly as before except scenes are *imagined* using the method to be evaluated. By applying the same procedure for all test scenes the number of correctly identified similar\dissimilar scene pairs can be treated as a success factor for image generation model. Figure 4.1 illustrates the testing procedure of two scenes generated for the two sets of descriptions for a single ground truth scene.

Note that other models (phrase-based SMT and MILP) considered throughout this report were trained on the training data that does not consist any of the test scenes required for the VP evaluation. In other words, all scenes used for testing by Lin and Parikh (2015) were assured to be in the testing sets of all the other models.

Linguistic general form	VP score
Main dependencies (lemmas)	41.07
Updated	42.41
Updated + combinations	**42.79**
Updated + singleton	40.96

Table 4.1: Comparison of different phrase-based SMT models used in terms of the generated image's quality. MILP solved using phrase-based SMT model from Section 3.1.1 and three improved models described in Section 3.1.2. Models evaluated using Visual Paraphrasing (VP) task (Section 4.1). Results suggest that *Updated + combinations* model is the best for generating images, however it is not significantly better than other alternatives.

4.2 Automatic Evaluation

In this section, models introduced in this report are evaluated and compared using automatic evaluation metric defined in Section 4.1. Visual Paraphrasing task is applied to measure how well two images generated correspond to the same scene. Training and testing sets are as specified in Section 4.1. Three main evaluation comparisons are performed.

Recall the results for different phrase-based SMT models in terms of F_1 scores in Table 3.2. It is difficult to distinguish which phrase-based SMT model is the best. Note, however, that the ultimate task is not to have a good working translator from descriptions to visual sentences, but to generate coherent images. Thus, MILP defined in Section 3.2 is solved using different phrase-based SMT models, and the resulting generated images are compared to provide a better understanding of which phrase-based SMT model is the best. Table 4.1 summarises the results. *Updated + combinations* model is shown to perform a little better than the other, thus we will use this phrase-based SMT model for the final comparison with other baselines described in Section 4.1. Human based performance analysis is additionally performed in Section 4.3 to support the results.

Next, the comparison of the MILP with different sets of constraints is executed to support the ideas in Section 3.2.3. The results are presented in Table 4.2. The results do not provide any useful insight as the Visual Paraphrasing task is too weak to distinguish between models with different sets of constraint in their MILP definitions. This is mainly because constraint removal does not make two scenes less or more similar to each other. For example, if prior belief positions are not considered, both images will have most of their clip arts placed in the upper top corner. Clearly, such set up would not be considered as a coherent image by human, however VP task only checks if two images are similar which is likely to be the case since clip arts in both images are placed in one specific corner. Consequently, human based performance analysis is performed in Section 4.3 to provide an intuition why some particular constraints in Section 3.2.3 are important for coherent image generation.

Finally, the *Full-MILP* which uses *Updated + combinations* model for translating de-

Model	VP score
MILP-f-g-only	42.74
MILP-g-h-only	42.58
MILP-f-h-only	43.13
Full-MILP	42.79
MILP-except-*above-below*	42.71
MILP-except-*on-surrounds*	42.87
MILP-except-*close-near-opposite*	42.84
MILP-except-*same-infront-behind*	42.83

Table 4.2: Comparison of MILP with different sets of constraints. f, g, h correspond to prior belief position, main, and prior belief relative position constraints respectively. Thus, models with the subset of these constraints do not consider the remaining ones. Similarly, models which include *except* do not consider subset of defined relations. *Full-MILP* defines the MILP model which considers all constraints specified in Section 3.2.3. The scores suggest that Visual Paraphrasing task is too weak to compare MILP's with different sets of constraints and that the human based performance analysis is needed to differentiate between them.

scriptions into visual sentences and full MILP definition to resolve clip art positions in the image is compared with other baseline models including:

GT ground truth baseline uses the original scenes for which the same linguistic descriptions were created by Mechanical Turk workers. Since all of these scenes match the sentences well, the best we can expect from any other approach is to tie with the ground truth.

BoW the bag-of-words baseline selects the most similar scene as output from the training dataset based on the bag-of-words representation for the input description. Description similarity is calculated using *tf-idf* score, so that matched infrequent words are more important. Using this representation, the scene which description is the most similar for the given sentence is retrieved from the training dataset. Note that scenes themselves were not used as a selection criteria.

Random a random scene is picked from the training dataset as an output for the input description.

Full-CRF model built by Zitnick et al. (2013) using the same Abstract Scenes Dataset. In brief, scenes are generated by setting up a Conditional Random Field (CRF) model (Lafferty et al., 2001) where each node corresponds to an object, and edges correspond to their relations. Potentials of the CRF model absolute and relative positions, occurrence and co-occurrence, and attributes of the objects. Potentials are determined using tuples extracted from the training set descriptions and visual features trivially identified from the scenes. Novel scene is then generated by sampling 30,000 potential scene set ups from the CRF and selecting the most probable one. Details can be found in Zitnick et al. (2013). Note that this model is the only baseline

Model	Novel	VP score
Random	No	33.40
BoW	No	34.80
Full-MILP	Yes	42.79
Full-CRF	Yes	43.36
GT	-	78.84

Table 4.3: Comparison of *Full-MILP* model with other baselines introduced in Section 4.1. Our model significantly outperforms both *Random* and *BoW* models, however, it generates slightly worse images compared to *Full-CRF* model (Zitnick et al., 2013) according to VP task. Note that ground truth *GT* images were given the score of only 78.84% which shows how difficult it is to distinguish if two different scenes are representing similar situation.

which generates novel scenes.

Note that because all compared models provide[4] a coherent image the VP task is a reliable way to evaluate the performance. The results are summarised in Table 4.3. *Full-MILP* model significantly outperforms both *Random* and *BoW* models, however it is not considered any better than *Full-CRF* model by Zitnick et al. (2013). Considering that VP task model is trained on images generated by *Full-CRF* model, i.e. *Full-CRF* is treated as a reliable method, it might be the case that our *Full-MILP* model is bounded by the performance of *Full-CRF*. Thus, human based performance analysis is additionally performed to review the performance of *Full-MILP* and other baselines, including *Full-CRF*.

4.3 Human Based Performance Analysis

It was noticed, in the previous section, that Visual Paraphrasing task is not always a fully reliable way to evaluate the quality of generated images. Therefore, human based performance analysis for all automatically executed comparisons is additionally completed to provide a full insight into the quality of the images.

First, the differences between the influence of phrase-based SMT models (Table 4.1) for the generated images are reviewed in Section 4.3.1.

Next, MILP definition is analysed by comparing the images generated using the subsets of MILP constraints. Concretely, models in Table 4.2 are explored by manually investigating the images. This step is presented in Section 4.3.2.

Finally, our full MILP model is compared with other baselines by exploring the nature of images generated by each of the models. This is presented in Section 4.3.3.

[4]Not all baselines generate a novel image. Some just retrieve one from the training dataset.

4.3.1 Phrase-based SMT Performance

In this section different phrase-based SMT models performance is analysed. Figure 4.2 presents multiple examples of the given sets of descriptions and the corresponding generated images for each model.

None of the models work either well or poor in all the situations, however there are trends which are easy to spot. *Main dependencies (lemmas)* model fails to capture most of the relations, therefore the generated images do not have a full information covered from the given descriptions. On the other hand, *Updated + singleton* model picks most of the objects in the given descriptions, however it does not always manages to capture relations between objects (as it considers many singletons) which results in clip arts being placed in prior belief positions. Finally, both *Updated* and *Updated + combinations* models perform similarly, however because the latter one captures the relations better (second and third examples) the *Updated + combinations* model is slightly preferred by both human and automatic evaluation.

In conclusion, all four phrase-based SMT models do indeed perform similarly with each model being better than the others in some specific situations. Most of the errors occur by using all four models, however because *Main dependencies (lemmas)* model fails to pick most of the objects and relations in the descriptions whereas *Updated + singleton* model captures too many singleton dependencies other two model are preferred. Hence, *Updated + combinations* model is considered as the best model and is used for comparison with other image generation baselines.

4.3.2 MILP Analysis

In this section, MILP definition is analysed by comparing images generated by the fully defined MILP and number of MILPs which only have a subset of constraints introduced in Section 3.2.3. The aim of such analysis is to support the design decisions made.

First, the importance of each class of constraints is illustrated in Figure 4.3. Here, images are generated for the multiple example descriptions using the full MILP definition and three MILP definitions which are each missing one class of constraints f, g, or h respectively. Recall that f, g, and h correspond to prior belief position, main, and prior belief relative position constraints.

As can be seen from the example images, all of the constraint classes are important. Without prior belief position constraints f all clip arts are positioned in the left top corner of the image. Even though the main (from visual dependencies) constraints are satisfied in this case, the images could not be considered as meaningful and coherent as it would be hard to understand what was happening in the scenes. Additionally, items which should be standing on the ground are instead flying in the sky which is not as expected. Note that this behaviour of MILP is due the fact that once constraints are satisfied when the lowest value feasible solution is retrieved. Left top corner of the image is the point where both x and y coordinates are 0. Therefore, since the main

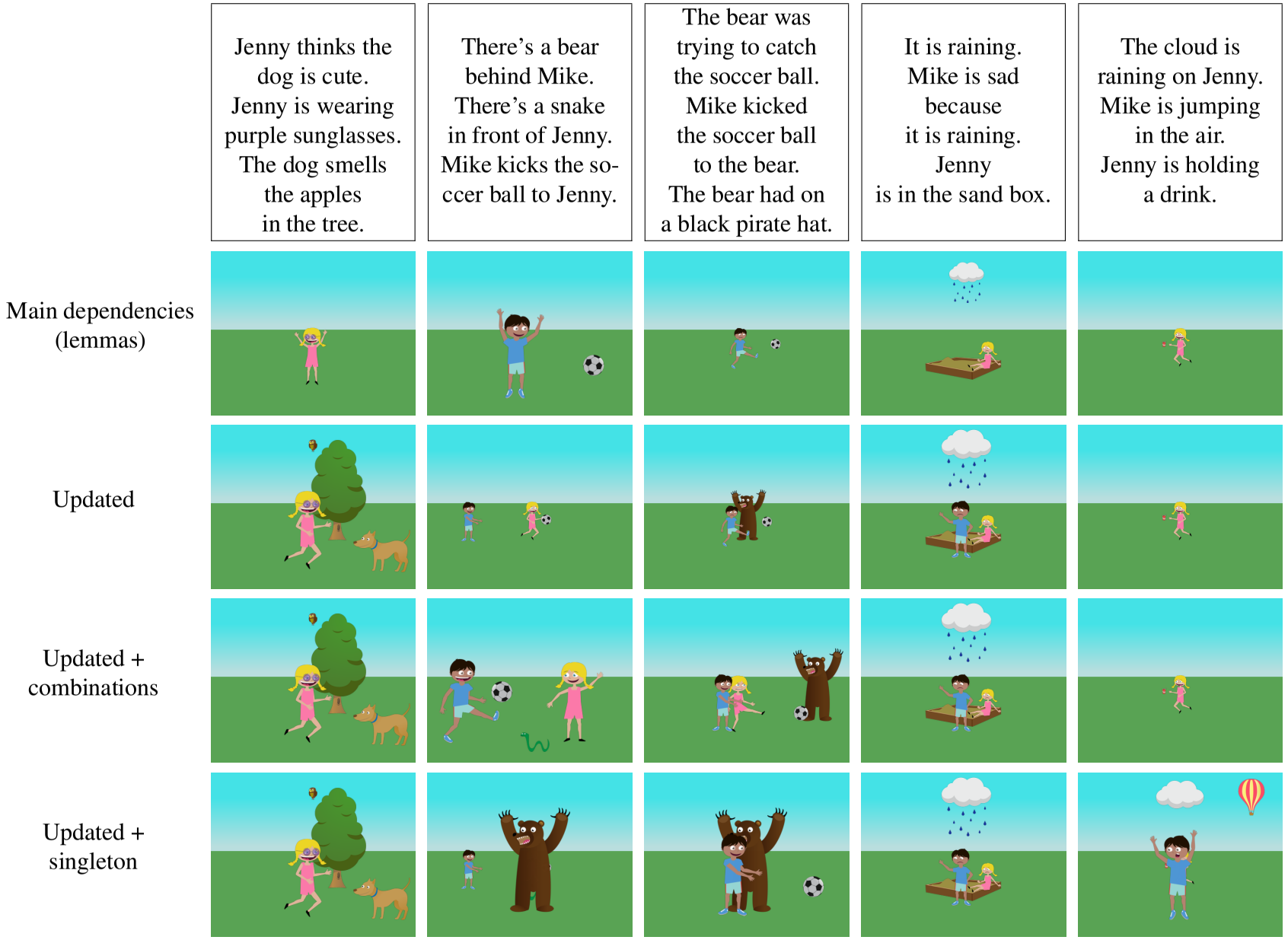


Figure 4.2: Examples of the scenes generated by the MILP using different phrase-based SMT model.

constraints do not have any absolute position constraints[5], MILP has no incentive to move any further from the left top corner.

If, on the other hand, main *g* constraints are completely removed, but the objects found in the visual sentences are still considered[6], then all clip arts are just positioned in their prior belief positions. Examples provided in Figure 4.3 illustrated multiple instances of this situation. Even though all the clip arts which are needed to be placed are present, their most likely to occur positions are not good enough to represent the relations in the linguistic descriptions.

Finally, prior belief relative position *h* constraints are also very important, however, their influence is not as clear as of other constraint classes. Consider the example descriptions in Figure 4.3 where the object pair mentioned has a constant similar relation. These include object pairs (*Mike*, *cap star*), (*Mike*, *hat*), and (*Mike*, *Frisbee*). Note that for all of these object pairs the *Full-MILP* model managed to correctly put the corresponding hat clip art on top of boy's head and similarly position Frisbee in his hands (as required by the description). This effect is taken care of by prior belief relative position constraints, and without them none of these clip art pairs would have been correctly positioned (see corresponding *MILP-except-h* model examples).

Therefore, we have shown that all constraint classes are very important in image generation task. Next, we analyse the importance of different relations (Table 2.1) within main *g* constraints. In all further experiments, both *g* and *h* constraints are considered when generating images unless stated otherwise.

Figure 4.4 presents the visual dependencies (translated by phrase-based SMT model) for the same example description sets as Figure 4.4 but the corresponding images are generated using different subsets of constraints. More precisely, the importance of different classes of relations (*above-below*, *close-near-opposite*, *on-surrounds*, *same-infront-behind*) is investigated by generating images using MILP which excludes one of these class constraints.

First, consider relations *above* and *below*. They are responsible for capturing clip art pairs where one clip art must be placed above or below the other. There are multiple clip art pairs (for example (*Mike, hat* $\Longleftrightarrow$ *hb0.0s.png, c.0s.png*), (*Mike, table* $\Longleftrightarrow$ *hb0.25s.png, p.9s.png*)) in the corresponding examples that hold one of these relations. Note that *Full-MILP* model in Figure 4.3 managed to position hat on boy's head and boy sitting on the table in the corresponding situations. However, if we now look at the *MILP-except-above-below* output we can see that boy is no longer sitting on the table. It shows the task of *above* and *below* relations. Note that in the third example the hat is still on the boy's head even though *above* and *below* relations were removed. This is because of *h* constraints applied.

Next, consider relations *close*, *near*, and *opposite*[7]. These relations are one of the most often appearing, thus their effect and importance is obvious in the example images.

[5]Visual dependencies represent relations between pairs of clip arts. These, however, do not specify anything about the absolute position of any of the clip arts.

[6]Otherwise images would be empty.

[7]Note that *closef*, *nearf* and *oppositef* are also included in this class.

The relation *on* and *surrounds*, on the other hand, are less often appearing, however still important. Consider the last example in Figure 4.4. Both hamburger and a pie, and table are coupled with *on* relation. Once this relation is removed from consideration a hamburger and a pie are no longer placed on the table but instead moved into their prior belief positions that happened to be close to the table.

Finally, one of the z relations *same*, *infront* or *behind* are always present in the combined[8] visual dependencies, therefore each example for *MILP-except-same-infront-behind* model in Figure 4.4 illustrates the effect of not having these relations. Note that h constraints are introduced for each z relation and since there are none in this situation the h constraints are also not present. The effect of z relations is easy to be seen from the examples. Without them depth relations are ignored which decreases the quality of the generated images. For example, the size of boy and hat clip arts are no longer consistent (they should be at the same depth in order for the hat to be of proportional size to fit on boy's head), similarly sizes of a pie and hamburger are disproportional to the size of the boy clip art (both are unrealistically big compared to the child). Thus, z relation constraints are important in coherent image generation task.

In this section, the significance of each of the proposed constraints in Section 3.2.3 were illustrated by examining multiple different generated images. We showed that all of the constraints provided are important in different various situations and the quality of the images are better with all of them applied.

4.3.3 Baseline Comparison

In this section the *Full-MILP* model which uses *Updated + combinations* phrase-based SMT model to translate given linguistic description to visual sentences and full MILP definition to resolve the variable values for all clip arts in visual sentences is compared with other baseline models. As evaluated automatically, *Full-MILP* model significantly outperformed both *Random* and *BoW* models, however, it had a very similar score as *Full-CRF* model (Zitnick et al., 2013). Therefore, this section aims to evaluate these models using human judgements. We do not manually analyse *Random* method outputs as they are believed to be poorly matching the given descriptions[9].

Figure 4.5 presents multiple example descriptions and the corresponding generated coherent images by *Full-MILP* and *Full-CRF* models. Ground truth *GT* scenes are added for reference. Images retrieved using bag-of-words model are included for comparison. Even though the images generated by both *Full-MILP* and *Full-CRF* models are consistent with the given descriptions and ground truth image, they are still quite different confirming the fact that there are multiple imaginations for the same set of sentences. Couple of trends were spotted by reviewing the images generated by both methods. First, *Full-MILP* model does not add clip arts that did not appear in the given

[8] Not singleton visual dependencies are called combined i.e. relation between two clip arts.

[9] Abstract Scenes Dataset includes more than 10,000 various scenes each of which represents a different situation. It is unlikely that randomly retrieved image will visually describe the given linguistic sentences in any sensible way.

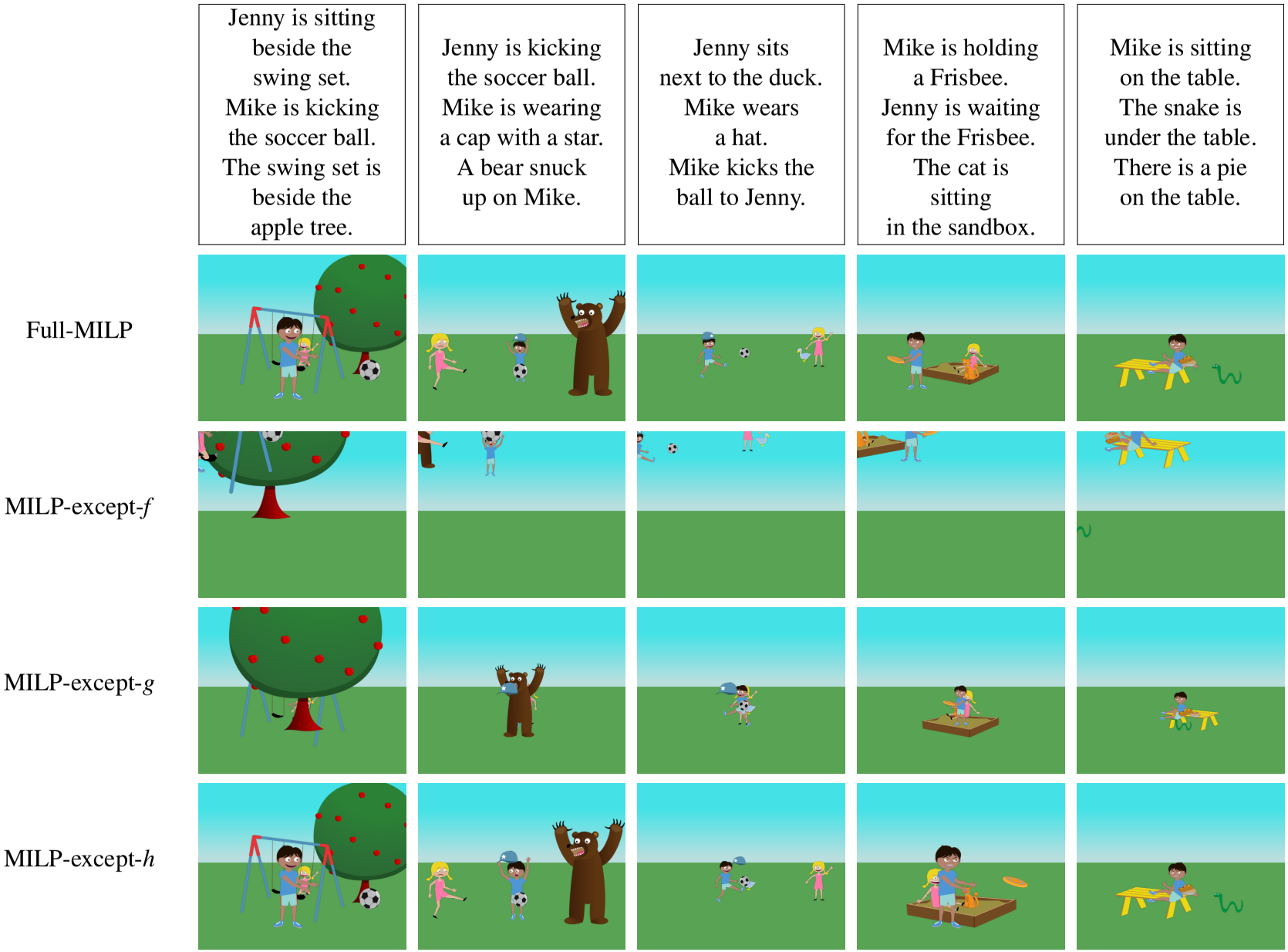


Figure 4.3: Examples of the scenes generated by the MILP as defined in Section 3.2.3 and MILPs that exclude either prior belief position (f), main (g), or prior belief relative position (h) constraints.

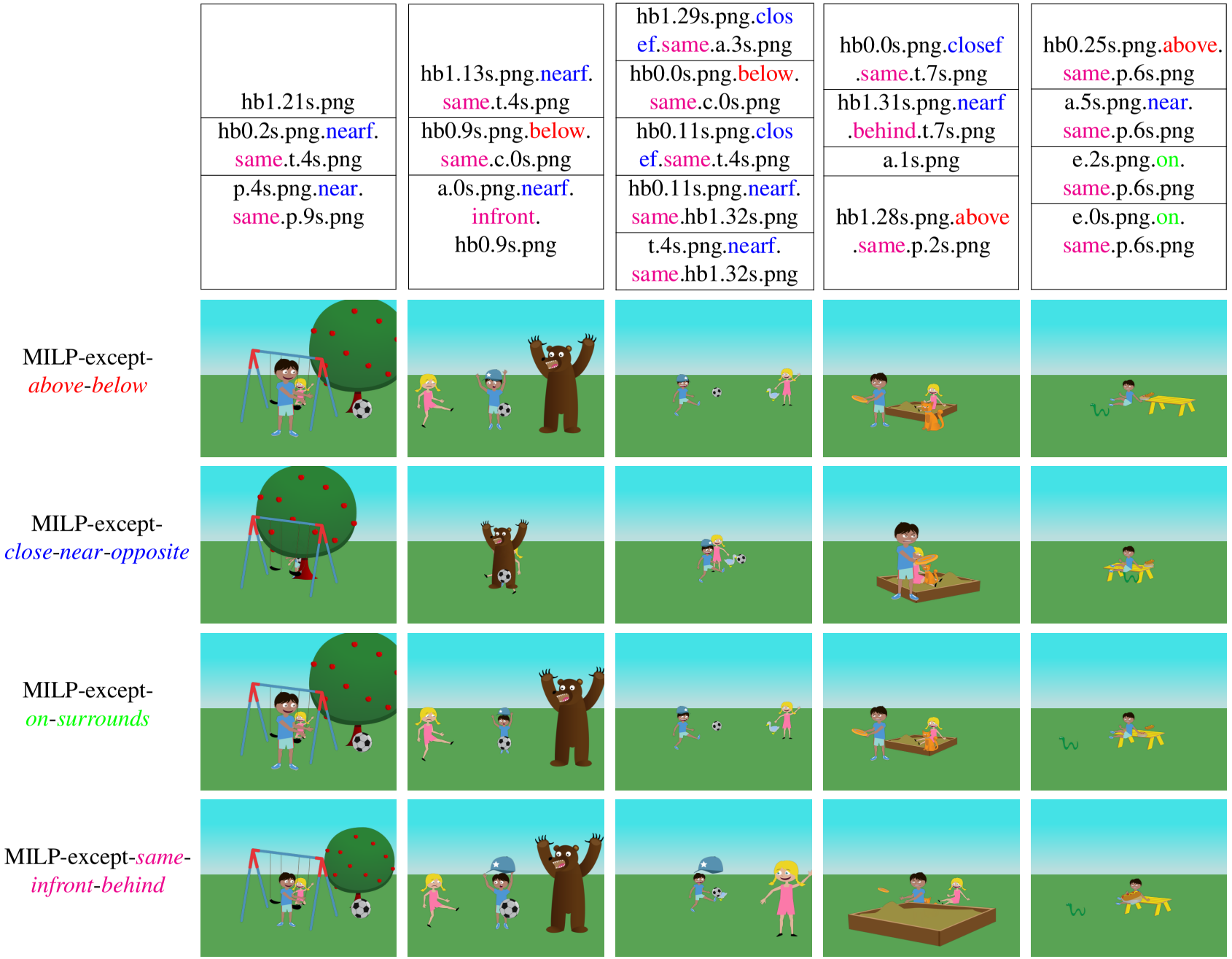


Figure 4.4: Examples of the scenes generated by the MILP as defined in Section 3.2.3 but excluding some of the relations from visual dependencies. Visual sentences are provided instead of linguistic descriptions to better understand what specific clip arts are affected by relation removal.

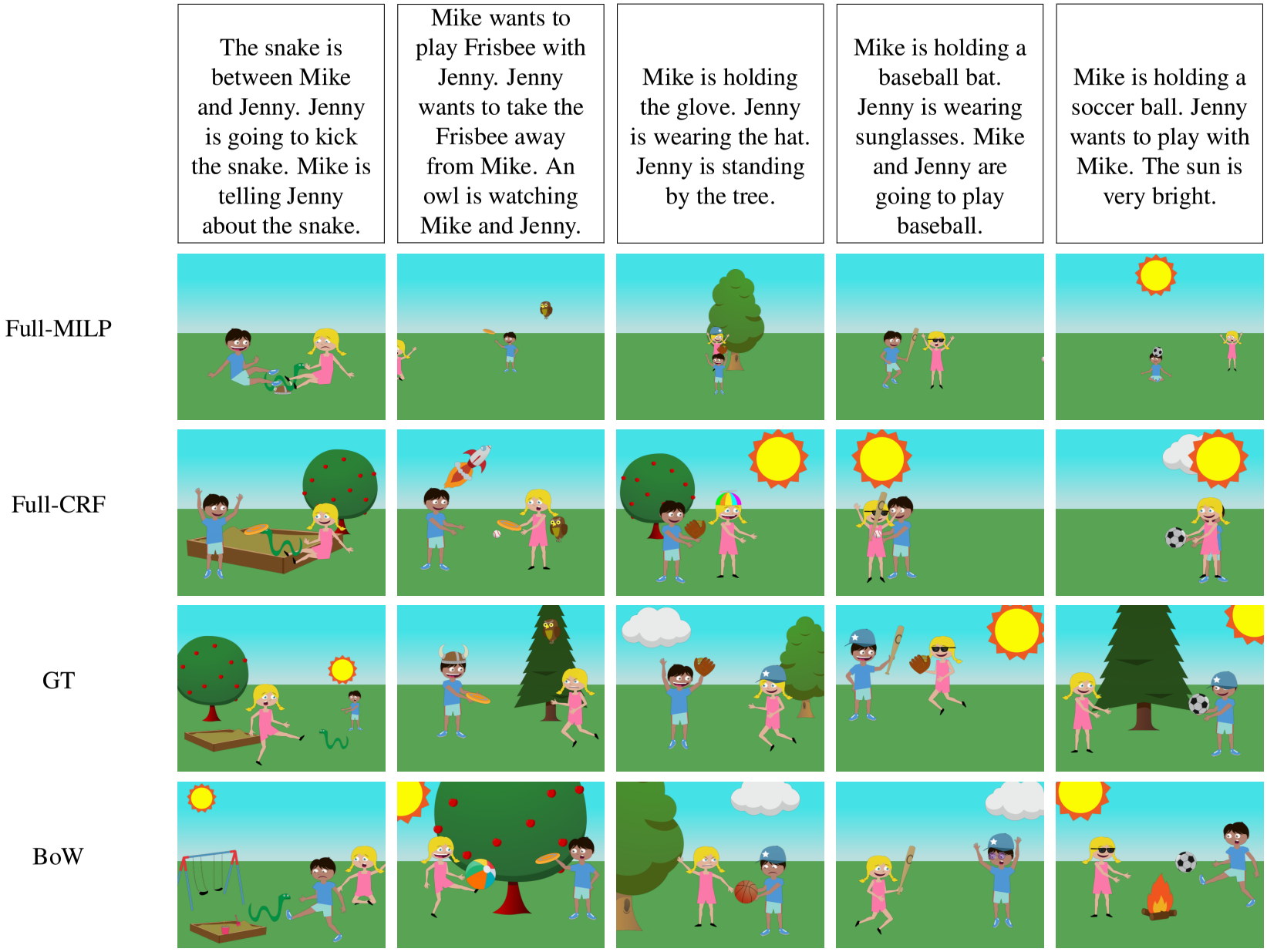


Figure 4.5: Examples of coherent scenes generated by both, *Full-MILP* and *Full-CRF* methods. Images retrieved using bag-of-words *BoW* method are shown for comparison. Ground truth *GT* scenes are included for the reference.

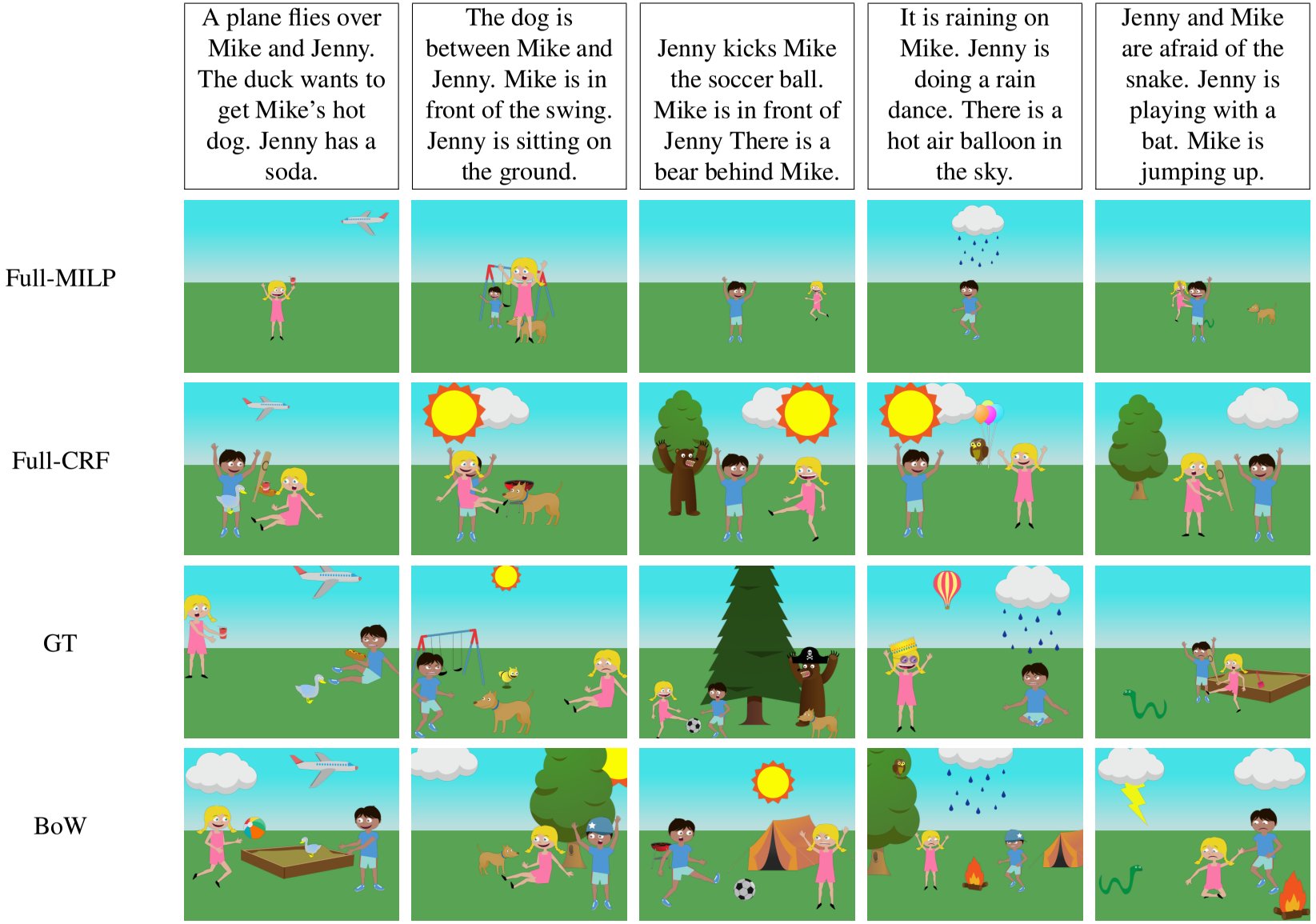


Figure 4.6: Examples of bad scenes generated by both, *Full-MILP* and *Full-CRF* methods. Images retrieved using bag-of-words *BoW* method are shown for comparison. Ground truth *GT* scenes are included for the reference.

sentences[10], whereas *Full-CRF* model does sample additional clip arts to generate images which better fit the nature of Abstract Scenes Dataset.[11] Such design decision makes *Full-MILP* images less crowded, sometimes unspecific, however it makes sure that given descriptions are visually illustrated. *Full-CRF* images, on the other hand, also visually represent the linguistic sentences well, however, it contains more clip arts to show a *bigger picture*. We argue that both methods are acceptable, since they both do the required job well, and it is the matter of need (full imagined scene or specifically illustrated descriptions) of future applications to select the appropriate method.

Another trend recognised in *Full-CRF* images is the prioritisation of larger scale of clip arts (or, in other words, smaller depth value) whereas *Full-MILP* model does not yet consider that. Even if images generated by *Full-MILP* model are coherent with given descriptions, they usually look empty just because the clip arts are of the smaller scale. Making them bigger would produce the illusion of better filled space in the image and hopefully more acceptable images for human eye. Such experimentation is part of our future work.

Not all of the images generated by *Full-MILP* and *Full-CRF* models are fully coherent. There is also a number of bad examples and some of these are shown in Figure 4.6. The main problems identified in the *Full-MILP* model are missing information about a specific linguistic sentence (or pair of objects) and an incorrect child's pose or facial expression. Both of these main problems arise from the phrase-based SMT model failures and/or finding the suitable child representing clip art. Note that if phrase-based SMT model fails to translate the given description, then all the information from that sentence is going to be missing in the imagined scene. This usually happens with a little bit more complicated and unique linguistic descriptions such as *The duck want to get Mike's hot dog* or *Jenny is doing a rain dance*. Having an even larger dataset could solve such problem. Otherwise, other methods should be considered to reliably translate linguistic descriptions into visual sentences as part of the future work. As mentioned, another noticeable problem is the incorrect child's pose or facial expression. Even though phrase-based SMT is mostly responsible for that, the child's pose and facial expression selection procedure could be further improved to also use relations with other clip arts to better decide in what pose the child is and what is his/her expression. Other problem is easily recognised for sentences which include conjunction between objects (for example *Mike and Jenny*). Note that current *Updated + combinations* phrase-based SMT model does not consider *conj* dependencies between words as the ones holding information about predicate objects. Thus, by design, all instances with conjugative objects will be missed. It is part of our future work to include *conj* dependencies into consideration.

On the other hand, problems in *Full-CRF* model are not exactly the same. There is lower number of missing clip arts in the *Full-CRF* scenes as more clip arts are included in the images in general. However, some of them are still missing and can be seen in Figures 4.6 and 4.7. Another problem with *Full-CRF* model is inability to reliably position clip arts which have a correlated relative position in the training data. Prior

[10] All clip arts were selected from visual sentences only, as described in Section 3.2.1.

[11] Recall that each image in the Abstract Scenes Dataset consists of *at least* 6 clip arts.

belief relative position constraints h solved this problem in *Full-MILP* method better than the approach by *Full-CRF*.

There are also multiple examples where images generated by *Full-MILP* are better than by *Full-CRF* (Figure 4.7) and vice-versa (Figure 4.8). The problems in both sets of examples are similar to the ones described before.

It might look odd that the third *Full-MILP* example in Figure 4.7 is considered correct even with the *owl* not quite positioned in the *tree*. Recall that MILP consider the bounding box of the clip arts when positioning them with respect to either *on* or *surrounds* relations. This was a simplification over the fact that dealing with bounding boxes is easier than mathematically defining the complete shape of the object boundaries within the clip art. Therefore, in the example above, an *owl* does indeed lie in the bounding box of the *tree*. As part of the future work more precise techniques would be considered.[12]

During the baseline model comparison, *BoW* model has not yet been reviewed. By looking at the examples in Figures 4.5, 4.6, 4.7, and 4.8 it is easy to convince the human judge that images retrieved using bag-of-words do not imagine the given descriptions well. Even if most of the relations in the descriptions are also present in the scenes, then it is most likely that the objects are mixed up. Consider a single example from the Figure 4.5, fourth description. Note that both relations *holding a baseball bat* and *wearing sunglasses* are present in the scene, however, the subjects are incorrect. *BoW* model is obsessed about finding the largest number of matching keywords in the descriptions, however, it does not mean that the relations between objects are consistent and that is there the *BoW* model fails to compete with both *Full-MILP* and *Full-CRF*.

In this section the *Full-MILP* model was analysed in more depth and compared with other baselines. It was straightforward to see that our developed model outperforms *Random* and *BoW* models, however a detailed analysis was done to compare it with *Full-CRF* model proposed by Zitnick et al. (2013). We have shown that each model have its own advantages and disadvantages.

[12] Such as learning prior belief relative positions for pair of objects which are coupled with relation *on* or *surrounds* and using that to position clip art (for example, an *owl*) on the most likely *on**surrounds* position of the other clip art (for example, a *tree*).

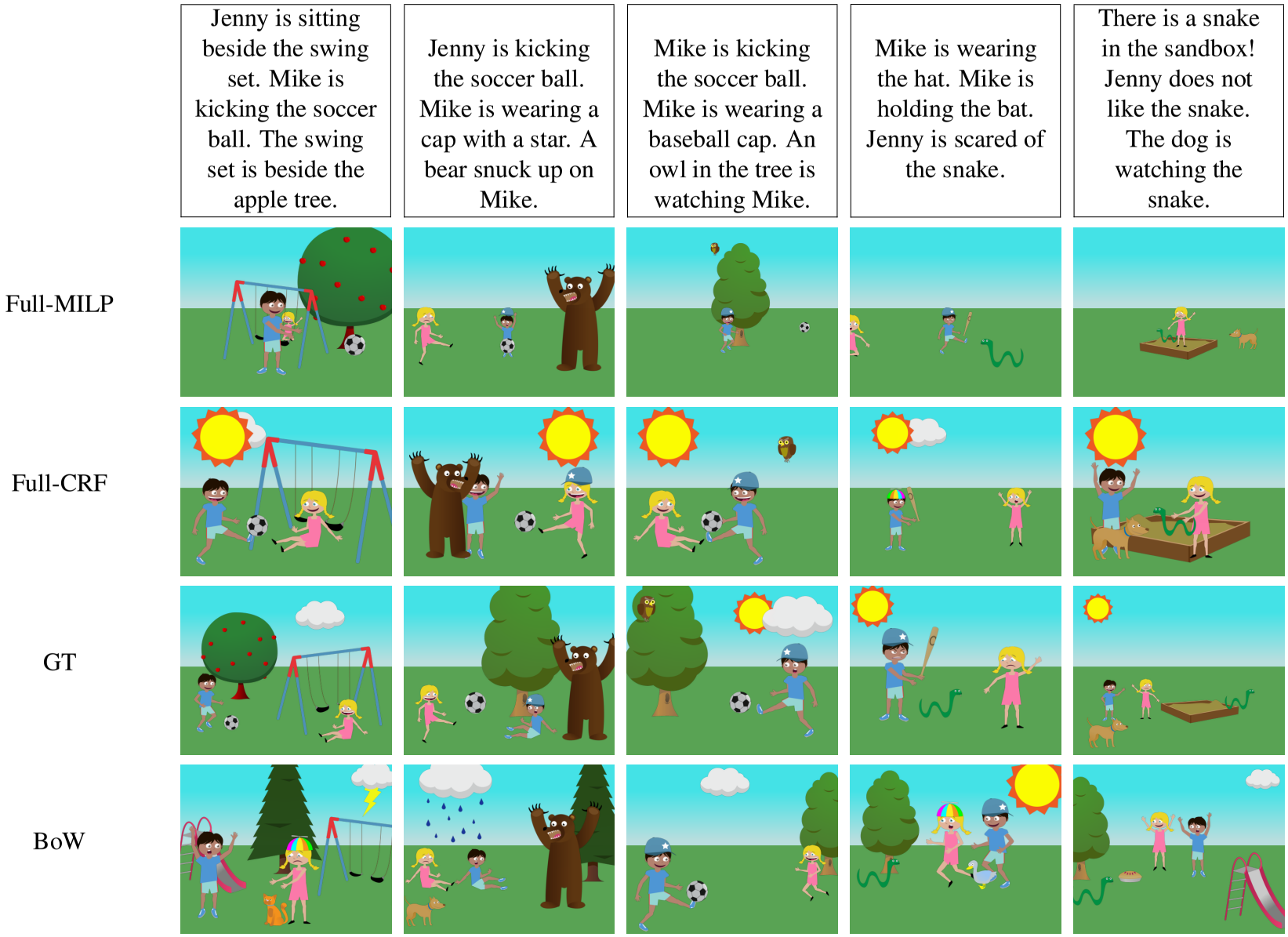


Figure 4.7: Examples of the scenes generated by *Full-MILP* that are better comparing to the corresponding scenes by *Full-CRF* method. Images retrieved using bag-of-words *BoW* method are shown for comparison. Ground truth *GT* scenes are included for the reference.

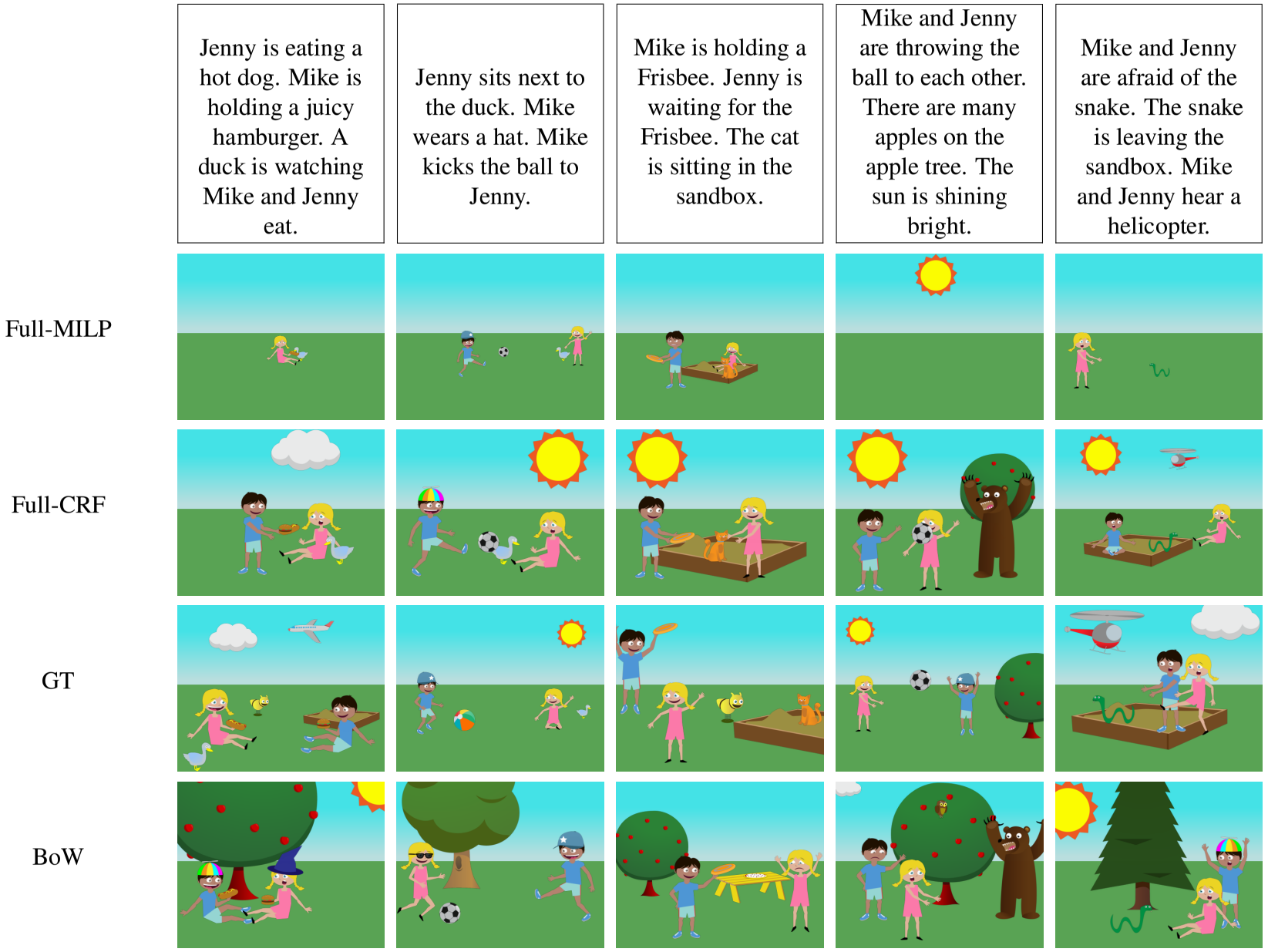


Figure 4.8: Examples of the scenes generated by *Full-MILP* that are worse comparing to the corresponding scenes by *Full-CRF* method. Images retrieved using bag-of-words *BoW* method are shown for comparison. Ground truth *GT* scenes are included for the reference.

Chapter 5

Conclusions

This report described a project done with the main goal to generate clip art images, coherent with textual descriptions given. The Abstract Scenes Dataset (Zitnick and Parikh, 2013) containing simplistic clip art images with a set of high quality visual features and corresponding human annotated descriptions played a crucial role in the project. It allowed us to focus on the core problem of the semantic scene understanding, while avoiding problems arising from the image feature extraction techniques and simplifying the task, so that instead of *drawing* the image it is rendered by conveniently specifying clip arts and their variable values.

In this chapter, the final comments are made to summarise the main contributions of the project in Section 5.1, and give the final remarks as well as possible further research in Section 5.2.

5.1 Summary of Contributions

To summarise the main contributions of this project, they are outlined as follows.

First, the two step methodology was defined to generate novel images from the given linguistic descriptions. The first step aimed to translate textual descriptions into visual sentences using phrase-based Statistical Machine Translation model proposed by Ortiz et al. (2015). Multiple modifications and improvements were made to the structure and creation of the parallel corpus needed to train this model to increase the phrase-based SMT model performance. These improvements were outlined in Section 3.1.

Most of the work was then devoted to the second step of the methodology which had a goal of identifying the clip arts which have to be placed in the resulting image, and defining Mixed Integer Linear Program, the solution of which would provide all the information needed to render the clip art image. This information included positions x and y, depth z and facing direction d variable values for each clip art which was selected to be included in the image. MILP was set in such a way that relations between clip arts, as defined by visual sentences, are satisfied. Additionally, prior belief clip art positions and prior belief relative positions for clip art pairs were learned from

the training dataset and included into MILP definition to improve the quality of the resulting images. Design of the second step of the methodology was presented in Section 3.2.

The experimentation was then performed to find the final model set up as a combination of the best phrase-based SMT and MILP models. This final model was compared with other three baselines, strong novel approach by Zitnick et al. (2013) and two straightforward image retrieval methods, using automatic evaluation and human based performance analysis. The application of the Visual Paraphrasing task was proposed as a reliable metric for evaluating clip art images. The experimentation and evaluation were described in detail in Chapter 4.

Our model was proved to significantly outperform both image retrieval methods and be competitive with the novel Zitnick et al. (2013) approach.

5.2 Final Remarks

Even though the model proposed in this report was competitive with a strong novel approach by Zitnick et al. (2013) there are multiple areas which could be improved or further experimented with.

First, the phrase-based Statistical Machine Translation model could be further improved by including even more predicate objects into the parallel corpus. Note that problems were spotted due to the missing *conj* dependency in the consideration of predicate objects as pointed out in Section 4.3.3. Another major problem with the phrase-based SMT model is empty translation. There were textual description instances which were translated into an empty visual sentences due to which all information covered by textual sentence was missing. It would be desired to avoid such cases by performing additional search in the given descriptions if no output was produced by the SMT model.

Multiple additional improvements could be considered in the second part of the image generation methodology, more specifically, MILP definition. Note that no constraint importance experiments were performed due the course of the project. This is a complicated task, however, it might help to increase the quality for images generated from contradictory visual sentences. Other issues which were already mentioned in the report include prioritising the image set ups where clip arts are in the closer depth to make a more coherent image, and considering prior belief relative positions for *on* and *surrounds* relations, so that the problem with an *owl* in the third example image of the *Full-MILP* model in Figure 4.7 is removed.

Note that, since Mixed Integer Linear Programming approach was selected for the second step of the methodology, there were some problems converting the constraints into the linear equations. Linear Programming is not the only approach which could have been used here. Quadratic Programming, on the other hand, could have easily handled the constraints which we had to approximate. However, due to more complex definition, the amount of time needed to find the solution could increase significantly, which

in this case merely took seconds. Regardless, it would be beneficial to experiment with different models to either prove that linear models are the best or find some better approach.

In this section, multiple suggestions for the future work were given considering the findings during the course of the project. The area of semantic scene understanding and image generation is still not widely researched, thus many possible further research paths exist.

Appendix A

Absolute Value Removal

In some cases constraint and objective functions include absolute value over the set of parameters. This chapter gives an intuition on how they can be translated into purely linear functions and equations.

To begin with, problem including absolute value function within the objective function of the form

$$\underset{x}{\text{minimize}} \quad |f(x)| + g(x)$$

is translated into the following problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f^A + g(x) \\ \text{subject to} \quad & -f^A \leq f(x) \leq f^A \\ & f^A \geq 0 \end{aligned}$$

where f^A is a new positive variable. Same procedure is applied for all absolute value functions in the objective. If absolute values appear in the constraints, then the conversion procedure is more complicated, since there are three different cases to it:

- Constraint of the form $|f(x)| = g(x)$ can be translated into the linear constraints in the straightforward way:

$$f(x) = g(x)$$

$$-f(x) = g(x)$$

- Constraint of the form $|f(x)| \leq g(x)$ is translated into the following linear constraints:

$$f(x) \leq g(x)$$

$$-f(x) \leq g(x)$$

  note that for the absolute value of $f(x)$ to be lower than some maximum value $g(x)$, both $f(x)$ and $-f(x)$ must be lower than $g(x)$.

- Constraint of the form $|f(x)| \geq g(x)$ is translated into the following linear constraints:

$$f(x) \geq g(x) - M \times b$$

$$f(x) \leq M \times (1-b) - g(x)$$

where b is a new binary variable $b \in \{0,1\}$ and M is a big constant.

Bibliography

Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 663–676, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15548-0, 978-3-642-15548-2. URL http://dl.acm.org/citation.cfm?id=1886063.1886114.

Bob Coyne and Richard Sproat. WordsEye: An Automatic Text-to-scene Conversion System. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 487–496, New York, NY, USA, 2001. ACM. ISBN 1-58113-374-X. doi: 10.1145/383259.383316. URL http://doi.acm.org/10.1145/383259.383316.

Desmond Elliott and Frank Keller. Image Description using Visual Dependency Representations. In *EMNLP*, pages 1292–1302. ACL, 2013. ISBN 978-1-937284-97-8. URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#ElliottK13.

Michael Grubinger, Paul Clough, Henning Muller, and Thomas Deselaers. The IAPR TC-12 benchmark: a new evaluation resource for visual information systems. In *International Workshop OntoImage*, pages 13–23, 2006.

Fritz Heider and Marianne Simmel. An Experimental Study of Apparent Behavior. *The American Journal of Psychology*, 57(2):243–259, 1944.

Dhiraj Joshi, James Z. Wang, and Jia Li. The Story Picturing Engine—a System for Automatic Text Illustration. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2 (1):68–89, February 2006. ISSN 1551-6857. doi: 10.1145/1126004.1126008. URL http://doi.acm.org/10.1145/1126004.1126008.

R. Kiros, R. Salakhutdinov, and R. S. Zemel. Multimodal Neural Language Models . In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603, 2014.

Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075150. URL http://dx.doi.org/10.3115/1075096.1075150.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

Xiao Lin and Devi Parikh. Don't Just Listen, Use Your Imagination: Leveraging Visual Common Sense for Non-Visual Tasks. *CoRR*, abs/1502.06108, 2015. URL http://arxiv.org/abs/1502.06108.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Neural Information Processing Systems (NIPS)*, 2011.

Luis Gilberto Mateos Ortiz. A Phrase-based Statistical Machine Translation Model for Automatic Image Description, August 2014. MSc Thesis, The University of Edinburgh.

Luis Gilberto Mateos Ortiz, Clemens Wolff, and Mirella Lapata. Learning to Interpret and Describe Abstract Scenes. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1505–1515, 2015. URL http://aclweb.org/anthology/N/N15/N15-1174.pdf.

Ken Perlin and Athomas Goldberg. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 205–216, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237258. URL http://doi.acm.org/10.1145/237170.237258.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting Image Annotations Using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 139–147, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1866696.1866717.

C. Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual

abstraction. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016, Portland, Oregon, 2013.

C. Lawrewnce Zitnick, Devi Parik, and Lucy Vanderwende. Learning the visual interpretation of sentences. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pages 1681–1688, Sydney, Australia, 2013.