

Generating Captions for Clip Art Images using Neural Language Models

Lukas Dirzys

MInf Project (Part 1) Report

Master of Informatics
School of Informatics
University of Edinburgh

2015

Abstract

In this report we propose a method to generate human-like image descriptions using multimodal neural language 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 our models by selecting optimal word and visual features by consistently experimenting with different feature sets and evaluating them with help of BLEU and Meteor automatic metrics. We find that the object's occurrence and its distance to a person's hand and head features are the best in reliably capturing main concepts presented by the image. We compare models trained with our feature set with two baselines - trivial language model that does not consider any image information at all, and models built with features extracted using computer vision techniques. Our proposed models significantly outperform both baselines. Due to automatic evaluation limitations in the task of caption generation we also analyse the descriptions manually and confirm the results of automatic scores.

Acknowledgements

Special thanks to my supervisor, Prof. Mirella Lapata, for the interesting proposed project as well as for her consistent help during the course of the project.

I would also like to thank Xingxing Zhang for helping me in setting up the GPU cluster and installing packages required for the models described in Section 2.2 to be trained.

Table of Contents

1	Introduction	1
1.1	Main contributions	2
1.2	Outline	2
2	Methodology	5
2.1	Dataset	5
2.1.1	Abstract Scenes Dataset	5
2.1.2	Reasons selecting the dataset	6
2.2	Models	8
2.2.1	Log-Bilinear Model (LBL)	9
2.2.2	Multimodal Log-Bilinear Models	11
2.2.3	Description generation	14
2.3	Evaluation Techniques	14
2.3.1	(smoothed) BLEU	15
2.3.2	Meteor	16
2.3.3	Qualitative Evaluation	17
3	Experimental Set-up	19
3.1	Building the model	19
3.2	Feature selection	20
3.2.1	Word features	20
3.2.2	Visual features	22
4	Results	27
4.1	Automatic evaluation	27
4.2	Human based performance analysis	31
4.2.1	ConvNet features performance	31
4.2.2	Abstract Scenes features performance	33
4.2.3	Discussion of the analysis	38
5	Conclusions	43
5.1	Summary of Contributions	43
5.2	Final remarks	44
5.3	Future Work	45
	Appendix A Cleaning the data	47

Appendix Bibliography**51**

Chapter 1

Introduction

Image caption generation task is one of the most challenging areas of research in computer vision. Humans, unlike computers, are able to easily and quickly capture the semantically meaningful information visually imparted by the image. Even though there might be dozens of different objects and actions described in the scene humans are still able to understand what objects are the most important, understand their relationships and summarize what is happening in the image just by using couple of sentences or even words. As an example, a human described the image in Figure 1.1 as a situation of two kids throwing a beach ball. A human chose to capture the most notable aspect of the image while other ones, such as the standing duck, the flying plane or clothes children are wearing, were considered as a side concerns.

Computers, unfortunately, are not yet so good at this task. There are many problems, such as reliable object recognition, reliable image features extraction, and the discovery of the most semantically meaningful image specific information, that need to be fully solved in order to automatically generate fluent and relevant to the image descriptions.

The automated image description task is challenging, however further research can



Jenny and Mike are throwing a beach ball.

Figure 1.1: Sample image and its description.

help to better understand human behaviour in selecting the most important aspects of the image worth to be described, and if successfully solved, could lead to a number of helpful real life applications. For instance, some applications include image search engine that can find images to accompany a text query (Feng and Lapata (2013), Barnard et al. (2001)), image search system for searching images of a specific property (such as object's occurrence) in a private directory (Krizhevsky et al., 2012) or the system for personal photos annotation (Ramnath et al., 2014).

1.1 Main contributions

Throughout the course of this project, we were focusing on discovering the most semantically meaningful image features that could help building the models capable of creating fluent, grammatically correct and relevant to the scene descriptions. To avoid using noisy features extracted with a computer vision techniques, we decided to use the Abstract Scenes Dataset that includes 10,020 clip art images with a set of image features already prepared Zitnick and Parikh (2013). The datasets also consist of 60,396 corresponding descriptions. (smoothed) BLEU and Meteor metrics were used for automatic models evaluation.

We selected multimodal neural language models Kiros, Salakhutdinov, and Zemel (2014) as a main approach for building automatic caption generation models. The experimentation was then done to select word and, the most semantically meaningful, visual features so that the performance of multimodal neural language models is optimised.

We introduced two baseline methods for the task of image description generation, the simple language model and two multimodal neural language models trained with features extracted with a computer vision techniques. We then evaluated models trained using our selected Abstract Scenes feature subset against these two baselines. Finally, the manual, human based performance analysis was done to confirm the automatic evaluation scores given.

Our proposed models significantly outperformed both baseline models in terms of automatic evaluation and human based analysis.

1.2 Outline

The report is organized as follows.

Chapter 2 introduces the selected methodology for creating models capable of producing relevant captions. The selected dataset, models, and evaluation techniques are described in detail.

Chapter 3 uses methods presented in Chapter 2 to optimise models by selecting word and visual features as well as model parameters.

Chapter 4 evaluates and compares our proposed models with two baseline methods using automatic metrics and manual human analysis. The chapter also outlines the limitations of automatic evaluation by showing the scores for the performance of human annotators.

Chapter 5 summarises our report, contributions, and findings as well as it gives an insight into the next year project's plan and goals.

Chapter 2

Methodology

This chapter presents our methodology for modelling, analysing, and evaluating image descriptions as well as describing the data used for training the 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 that are capable of generating semantically meaningful captions.

In order to learn the semantics from the training image features and associated set of descriptions we use Multimodal Neural Language Models as proposed by Kiros, Salakhutdinov, and Zemel (2014). These models are presented in Section 2.2.

Finally, we discuss image description evaluation and our approach in Section 2.3 in more detail. We use two automatic evaluation metrics (smoothed) BLEU and Meteor which correlate well with human judgements (Elliott and Keller, 2014) in evaluating and comparing our models.

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 (images) that 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 creating a realistic scene using a collection of clip art pieces. Workers created scenes

¹The 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.

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, 24 of which are a person (7 different poses and 5 different facial expressions for each child - boy and a girl) and other 56 represented other objects. A simple background 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. At the next step, 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 worker wished to use proper names.

Subsequently, with the help of 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 that 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 with one short sentence. Every scene was described from 3 to 9 times resulting in 60,396 descriptions in total. Only these 10,020 scenes with associated 60,396 short captions were used throughout the course of this report. To increase the quality of responses only the workers from United States were selected.

Another part of the dataset includes a set of visual features extracted from the 10,020 scenes created by the Amazon's Mechanical Turk participants². These features are split into different feature sets which represent specific properties of the image such as the objects' occurrence, co-occurrence, absolute location in the image, etc. The features are discrete (i.e. they specify the occurrence of a property) or continuous (i.e. specifies the location of a particular property). Discrete features include a specific object's occurrence; two objects co-occurrence, the occurrence of a person's pose or facial expression, and absolute, and relative depth³ while continuous features consist of the object's absolute and relative locations and distances between objects in the scene and child's hand or head. An example of the scene with its associated features is shown in Figure 2.1.

2.1.2 Reasons selecting the dataset

There are a couple of reasons for working with the Abstract Scenes Dataset over the realistic photos. As introduced in Chapter 1, identifying the specific parts of an image, such as 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

²Note that these features are known to be correct since they are collected during the creation of the scene (for instance, it is known at what position in the scene the clip art piece was placed, how the piece was scaled and whether it was flipped).

³Depth in the Abstract Scenes Dataset is defined as a value 0 (close), 1 (middle) or 2 (far away).



Occurrence		Co-occurrence	
Sun	1	Helicopter-Cloud	0
Boy	1	Cloud-Sun	0
Girl	1	Sun-Boy	1
Table	0	OakTree-Girl	1
Snake	0	Grill-Table	0
...

Absolute Location		Relative Depth	
Sun	0.057397	Sun-OakTree_0	1
OakTree	0.882495	Sun-OakTree_1	0
Swing	0.000000	Sun-Boy_0	1
Boy	0.093186	Balloons-Girl_0	0
Helicopter	0.000000	Girl-Boy_0	1
...

Person pose or face		Distance to head	
Boy_Pose_Hold	1	boy-Sun	0.000001
Girl_Pose_Kick	1	boy-BaseballCap	0.985605
Girl_Smile	1	girl-BeachBall	0.031242
Boy_Surprise	1	girl-Balloons	0.000000
Boy_Pose_Wave	0	girl-Hotdog	0.000000
...

Figure 2.1: Sample features for the example scene. Note that object names are presented for a clearer representation of the features. In reality, models trained using these features are not aware of which piece of clip art the specific word, such as *sun*, refers to.

no good solution for it. However, 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 Scenes Dataset should be applicable to photo-realistic images. Therefore, with the use of the Abstract Scenes Dataset it is possible to avoid noise and other problems coming from image feature extraction using computer vision techniques with no side effects on the resulting generated captions.

Indeed one could argue that the number of photo-realistic scenes is huge and that they could be easily extracted from the internet or other sources, and probably even with plausible associated descriptions that could be used for training. The Abstract Scenes Dataset is relatively large⁴ 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)) that include millions of images and their descriptions and are directly scraped from the web would include noise and unnecessary, unrelated to the image information, and the descriptions might also be written by non-native English speakers.

Lastly, since the Abstract Scenes Dataset includes stories that are represented by the set of different scenes with their own sets of different descriptions, semantically meaningful information in the scene can be explored better. By knowing the most important semantic information in the scene, we are given the advantage of generating captions that capture the most significant actions and properties in the image.

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

2.2 Models

Neural networks first introduced in mathematical and algorithmic form by Mcculloch and Pitts (1943) were used by Rosenblatt (1958) to create a notion of perceptron - a classification algorithm that makes a linear output prediction combining a set of weights for the feature vector. A huge amount of research has been made in the field since then and natural language models built with neural networks (multilayer perceptrons) are not an exception. Kiros, Salakhutdinov, and Zemel (2014) recently introduced multimodal neural language models which can be conditioned on other modalities and can be well used for the scene caption generation task where the scene is treated as an additional modality. They proposed two models, the modality-biased log-bilinear (MLBL-B) model and the factored 3-way log-bilinear (MLBL-F) model. Both models are based on a simple log-bilinear model (LBL) which relies on word representation vectors learned from a database of words by adding a model conditioned on image features learned from deep neural networks or extracted from other sources.

⁴As 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).

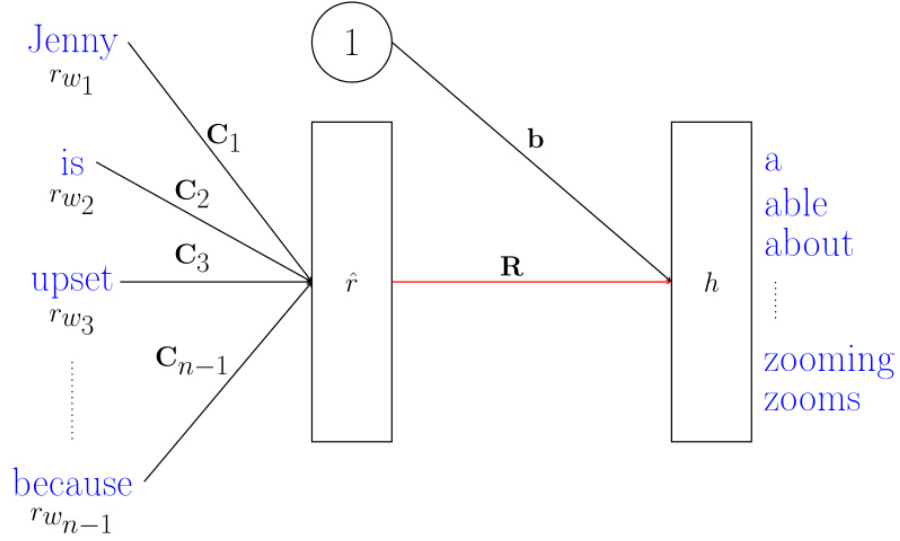


Figure 2.2: The Log-Bilinear Model (LBL)

The simplest, LBL, model is described in Section 2.2.1 in detail while both MLBL-B and MLBL-F extend the definition of the basic log-bilinear model in Section 2.2.2.

2.2.1 Log-Bilinear Model (LBL)

The Log-Bilinear Model (Mnih and Hinton, 2007) is a deterministic model based on a feed-forward neural network with a single linear hidden layer. The model operates on word representation vectors whose selection procedure is discussed in Section 3.2.1 and makes a linear prediction of the next word representation given the previous $n - 1$ words or in other words the context. This model is used as an absolute baseline since it does not consider any image information at all therefore generating most likely English sentence according to the training data.

Formally, each word w in the vocabulary which is the set of all training words is represented as a d -dimensional vector $\mathbf{r}_{w_i} \in \mathbb{R}^d$. For the context of size $n - 1$ let $(w_1, w_2, \dots, w_{n-1})$ be a tuple of $n - 1$ words. For each context word $w_i, i = 1, 2, \dots, n - 1$ consider a $d \times d$ context weight matrix \mathbf{C}_i initialized randomly from a zero mean Gaussian with a standard deviation of 0.01. The LBL model predicts the next word n representation $\hat{\mathbf{r}}$ as:

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}_i \mathbf{r}_{w_i}$$

The next step in LBL model is to score the predicted representation $\hat{\mathbf{r}}$ of w_i against the actual representation \mathbf{r}_w for every word in the vocabulary and select the one with the highest score. Let \mathbf{R} be the $k \times d$ matrix of all word representation vectors where k is the vocabulary size, and let $\mathbf{b} \in \mathbb{R}^k$ represent a bias vector with a bias b_w for each word

in the vocabulary. Word representation matrix⁵ \mathbf{R} and bias vector⁶ \mathbf{b} are considered as weights between the hidden layer and output layer in a neural network as shown in the Figure 2.2. The output layer uses a softmax activation function to calculate the vector \mathbf{h} of the conditional probabilities $P(w_n = w | w_{1:n-1})$ of w_n being a specific word w given the context words w_1, w_2, \dots, w_{n-1} . The softmax function is described in the Section 2.2.1.1. The next word w_n is selected as:

$$\begin{aligned} w_n &= \arg \max_i P(w_n = w_i | w_{1:n-1}) = \\ &= \arg \max_i P(w_n = w_i | \hat{\mathbf{r}}) = \\ &= \arg \max_i \sigma(\mathbf{R}\hat{\mathbf{r}} + \mathbf{b})_i, \\ & i = 1, 2, \dots, k \end{aligned}$$

The LBL model parameters can be learned with back propagation (Section 2.2.1.2). Once the model is built the caption for a new scene can be generated as described in Section 2.2.3.

2.2.1.1 Softmax activation function

Once the word representation matrix \mathbf{R} , the bias vector \mathbf{b} and the predicted next word representation $\hat{\mathbf{r}}$ are present it is not yet possible to tell which word is the most likely to be the next one given the context. The activation function is required in this step to map these inputs to the outputs of a single value for each word. Ideally, the outputs would correspond to probability distribution of being the next word amongst all possible words in the vocabulary. Such behaviour can be achieved with a use of softmax activation function (Bishop, 2006, p. 227-241) defined as:

$$\sigma(\mathbf{Z})_i = \frac{\exp(\mathbf{Z}_i)}{\sum_j \exp(\mathbf{Z}_j)}, i = 1, 2, \dots, k$$

where \mathbf{Z} is the input matrix for LBL model defined as $\mathbf{Z} = \mathbf{R}\hat{\mathbf{r}} + \mathbf{b}$. Since each value of the vector $\mathbf{h} = \sigma(\mathbf{Z})$ is strictly between zero and one and the sum of all these values is one, the softmax activation function does indeed produce a probability distribution over the final output values for each word in the vocabulary.

2.2.1.2 Learning algorithm

Learning the LBL model is simple and can be done with a straightforward application of backward propagation. Since the initialized word representation matrix \mathbf{R} , bias

⁵Can be initialized randomly or using pre-trained word embeddings that are explored in Section 3.2.1.

⁶Initialized to zeros.

vector \mathbf{b} and context weight matrices \mathbf{C}_i are present initial outputs can be calculated by forward propagating the network.

Training is done considering each training scene-caption pair in a row. As just mentioned, the predictive output (next word representation) is first calculated for each set of context words, and since the actual answer is given, the backward pass through the network is done and the gradients required for learning $\frac{\partial f}{\partial \mathbf{R}}$, $\frac{\partial f}{\partial \mathbf{C}_i}$ and $\frac{\partial f}{\partial \mathbf{b}}$ where f is a log-likelihood function $f = \log P(w_n | w_{1:n-1})$ are found. Network weights are then updated with respect to learning rate. Mnih and Hinton (2007) describe back propagation procedure in more detail.

Training for each scene-caption pair is repeated multiple times until the stopping criterion is satisfied (either maximum number, 100 for all further experiments, of iterations reached or performance on validation set is optimized). The validation set is created by splitting the training data into 80% (left for training) and 20% sets. All models were trained until the perplexity on the validation set no longer improved for 5 iterations.

2.2.2 Multimodal Log-Bilinear Models

The Modality-Biased Log-Bilinear (MLBL-B) and Factored 3-way Log-Bilinear (MLBL-F) models proposed by Kiros, Salakhutdinov, and Zemel (2014) with each words training tuple $(w_1, w_2, \dots, w_{n-1})$ takes an additional set of features $\mathbf{x} \in \mathbb{R}^s$ extracted from the image into the account while training. The selection procedure of visual features is described in Section 3.2.2.

2.2.2.1 Modality-Biased Log-Bilinear Model (MLBL-B)

The MLBL-B model is a simple extension of the LBL model discussed in the previous section. The model just adds an additive bias to the next predicted word representation $\hat{\mathbf{r}}$ to be computed as:

$$\hat{\mathbf{r}} = \left(\sum_{i=1}^{n-1} \mathbf{C}_i \mathbf{r}_{w_i} \right) + \mathbf{C}_s \mathbf{x}$$

where \mathbf{C}_s is a $d \times s$ image context matrix with d being the size of word representation vectors and s the number of image features. Given the predicted next word w_n representation $\hat{\mathbf{r}}$, the next step for scoring the predicted representation against real word representations and selecting the most likely next word w_n remains unchanged from the LBL model. The extension to the LBL model is shown in the Figure 2.3.

The model can be learned with standard back propagation⁷ and scene's description can be generated in the exact way as for LBL model.

⁷The difference between learning LBL and MLBL-B model is the additional weight matrix \mathbf{C}_s optimization which is similar to the procedure of training context weight matrices \mathbf{C}_i .

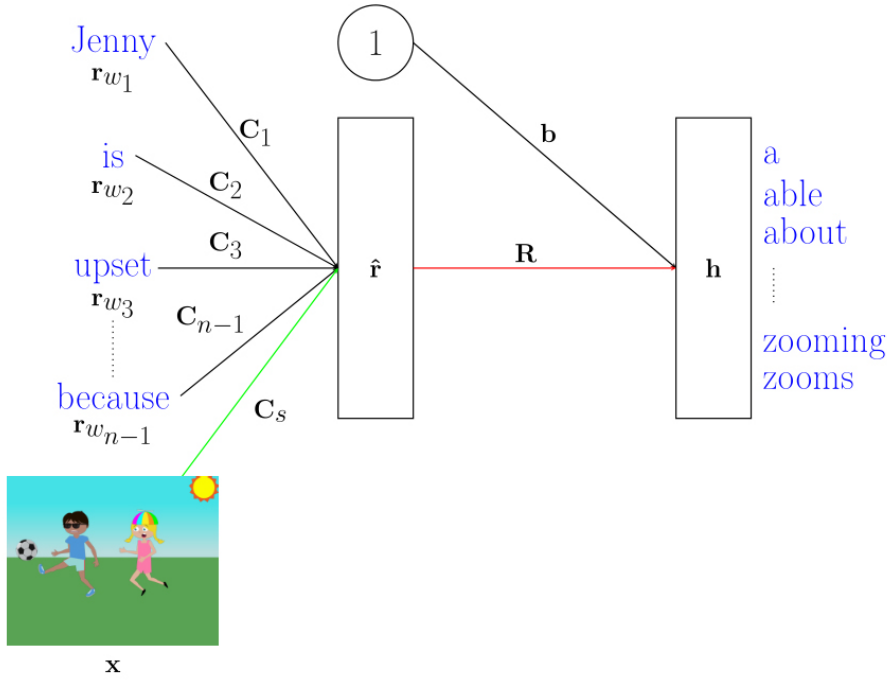


Figure 2.3: Modality-Biased Log-Bilinear Model (MLBL-B)

2.2.2.2 Factored 3-way Log-Bilinear Model (MLBL-F)

The MLBL-F model can use image features \mathbf{x} not only as an additive bias but also to gate the word representation matrix \mathbf{R} . By doing this, the word representation matrix becomes a tensor for which each image feature x can specify its own hidden to output weight matrix. The motivation of having scene specific word representation is that the model trying to predict the next word for the image containing a clip art of a cat and the example context words (*there, is, a*) would score \hat{r} higher for a noun *cat* than for other nouns such as *dog, owl* or *snake*.

Formally, suppose $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \dots, \mathbf{R}^{(s)}$ are $k \times d$ matrices specified for each of the feature $x_i, i = 1, 2, \dots, c$. The hidden to output weights for feature vector \mathbf{x} are found by:

$$\mathbf{R}_{\mathbf{x}} = \sum_{i=1}^c x^{(i)} \mathbf{R}^{(i)}$$

As Kiros, Salakhutdinov, and Zemel (2014) mention, the tensor \mathbf{R} requires $k \times d \times c$ parameters to be trained (since each feature has its own hidden to output weight matrix $\mathbf{R}^{(1)}$ of size $k \times d$). They propose to use Boltzmann machines (Memisevic and Hinton (2007), Alex Krizhevsky et al (2010)) works to factor \mathbf{R} into 3 matrices $\mathbf{W}_{f\hat{r}}, \mathbf{W}_{fx}$ and \mathbf{W}_{hf} such that:

$$\mathbf{R}_x^T = \mathbf{W}_{hf} \odot \delta(\mathbf{W}_{fx} \mathbf{x}) \odot \mathbf{W}_{f\hat{r}}$$

where $\delta(\cdot)$ denotes the matrix with its argument on the diagonal, \odot is the Hadamard

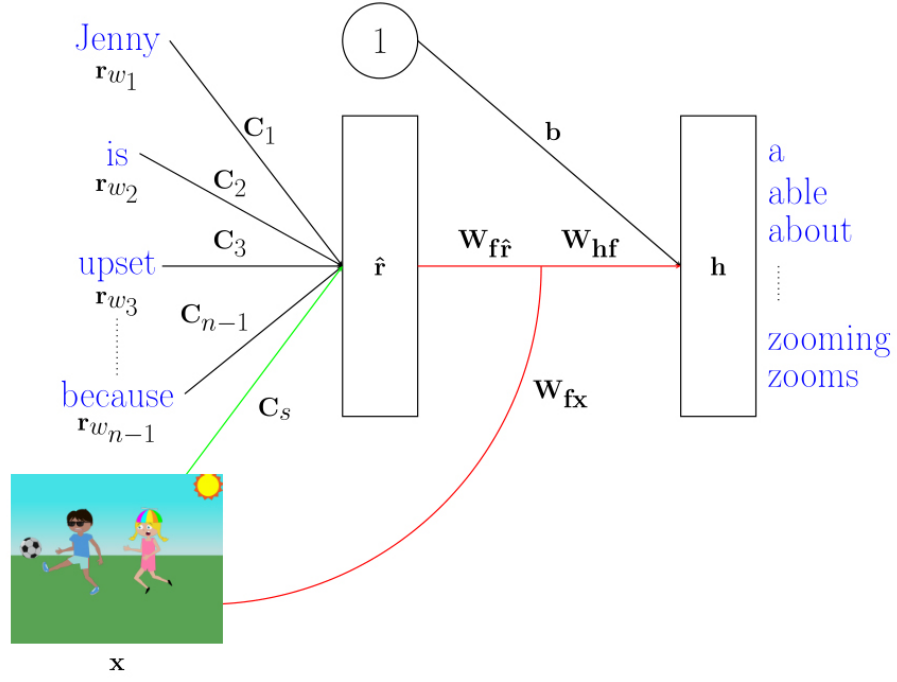


Figure 2.4: Factored 3-way Log-Bilinear Model (MLBL-F)

product⁸ and f is the parameter that denotes the number of factors these 3 matrices should be parametrized by. The predicted next word representation $\hat{\mathbf{f}}$ under this model is:

$$\hat{\mathbf{f}} = \left(\sum_{i=1}^{n-1} \mathbf{C}_i (\mathbf{W}_{hf} \mathbf{W}_{f\hat{\mathbf{f}}})^T_{w_i} \right) + \mathbf{C}_s \mathbf{x}$$

where $(\mathbf{W}_{hf} \mathbf{W}_{f\hat{\mathbf{f}}})^T_{w_i}$ is the column in $(\mathbf{W}_{hf} \mathbf{W}_{f\hat{\mathbf{f}}})^T$ for the word representation of w_i . Unlike MLBL-B, MLBL-F model does not use direct word representation vectors r_{w_i} to calculate the next word's predicted representation but updates them with respect to scene features \mathbf{x} .

Finally, the next word is selected as:

$$\begin{aligned} w_n &= \arg \max_i P(w_n = w_i | w_{1:n-1}) = \\ &= \arg \max_i P(w_n = w_i | \hat{\mathbf{f}}, \mathbf{x}) = \\ &= \arg \max_i \sigma(\mathbf{W}_{hf} ((\mathbf{W}_{f\hat{\mathbf{f}}} \hat{\mathbf{f}}) \odot (\mathbf{W}_{fx} \mathbf{x})) + \mathbf{b})_i, \\ & \quad i = 1, 2, \dots, k \end{aligned}$$

⁸Hadamard product between two matrices A and B of the same dimensions $m \times n$ is defined as $(A \odot B)_{ij} = (A)_{ij}(B)_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

where $(\mathbf{W}_{f\hat{\mathbf{r}}}\hat{\mathbf{r}}) \odot (\mathbf{W}_{f\mathbf{x}}\mathbf{x})$ are the factor outputs. Figure 2.4 illustrates and summarises the MLBL-F model. Comparing the next word selection procedure with MLBL-B model's approach it is easy to see that MLBL-F does not just use the predicted word representation directly but also makes sure the weights for each scene feature x can be learned in order to transform the original predicted word representation to be more scene specific.

Model training can be achieved using back propagation as with the previous models. New descriptions can be generated using the simple procedure described in the Section 2.2.3.

2.2.3 Description generation

Given one of the trained models described in the previous sections the description for new unseen scene can be generated as follows. Suppose we do have the required visual features extracted from the image (the visual features extraction techniques are represented in the Section 3.2.2). Also suppose we are given the initialization of the first $n - 1$ context words $w_{1:n-1}$. These are all initialized to the start token <start> in all further experiments.

The next word w_n (starting from $n = 1$) is computed as:

$$w_n = \arg \max_i P(w_n = w_i | w_{1:n-1}), i = 1, 2, \dots, k$$

and is appended to the initialization. This procedure is then repeated until one of the following conditions is met: either the end token <end> is generated or more than 50 words are already generated⁹ for the current caption.

2.3 Evaluation Techniques

The variability of fluent and accurate image descriptions is very high. Humans tend to describe the same image differently, since the scene usually includes many different components and separate actions that can be selected and interpreted in various ways. In addition, even when describing the same aspect, humans might use a million different word combinations to represent the identical meaning. The Abstract Scenes Dataset is not an exception. As shown in Table 4.1 of Chapter 4, human agreement is poorly scored by automatic measures (which are to be used) even when being aware that these descriptions are fluent and accurate.

These problems make evaluation task challenging, and even though the use of automatic evaluation might not represent the results well, we decided to use (smoothed)

⁹In practise, 50 words will never be generated, since the training sentences are relatively short. See Figure 3.1 for detailed training sentences length distribution.

BLEU¹⁰ (Papineni et al., 2002) and Meteor (Denkowski and Lavie, 2014) as Elliott and Keller (2014) estimated that these metrics at least correlate well with human judgements. It is still needed to perform qualitative tests to overcome a problem of low automatic evaluation interpretation.

The evaluation metrics mentioned above are described in more detail in the sections 2.3.1 and 2.3.2 below.

2.3.1 (smoothed) BLEU¹¹

The BLEU (Bilingual Evaluation Understudy) score introduced by (Papineni et al., 2002) is an automated metric to evaluate the performance of translation¹² systems which achieves high correlation with human judgements. The BLEU metric aims to score the predicted (candidate) sentences given a set of gold-standard human created sentences (references) by counting the proportion of n-grams in the candidate that occurs in any of the reference sentence. To avoid scoring candidates high which have a lot of words that are likely to appear in most of the references (such as *is*), BLEU clips the counts for each n-gram to the maximum number of occurrences in a single reference. Since in the current set-up short candidates with all words appearing in references would be scored high even though they do not capture the full meaning, BLEU introduces a brevity penalty for candidates shorter than a reference.

More specifically, the BLEU score for the candidate sentences is calculated by first computing precision scores p_n for all n up to length N (in our evaluation we are using $N = 4$). Parameter n represents what kind of n-gram matches between candidate and reference captions are considered. BLEU computes n-gram matches sentence by sentence. The precision score for every n is computed dividing the sum of clipped n-gram counts for each candidate sentence by the sum of total candidates n-grams in the test corpus:

$$p_n = \frac{\sum_{C \in \text{Candidates}} \sum_{n\text{-gram} \in C} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{C \in \text{Candidates}} \sum_{n\text{-gram} \in C} \text{Count}(n\text{-gram})}$$

Next, let c be the length of the candidate translation and r be the effective (reference sentence with the closest to candidate's length) reference corpus length. Brevity penalty BP is computed as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

¹⁰Calculated by using NIST's MultEval software: <ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a-20091001.tar.gz>

¹¹The terms (smoothed) BLEU and BLEU are used interchangeably in this report.

¹²The task of scene caption generation can be seen as a translation system between image and its corresponding description.

that penalizes candidates shorter than the effective reference. The geometric average of the n -gram precisions p_n is computed using positive weights w_n summing to one.

We use uniform weights $w_n = \frac{1}{N} = \frac{1}{4}$. Finally, BLEU score is calculated as:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^N w_n \log(p_n)\right) = BP \cdot \exp\left(\frac{1}{4} \sum_{n=1}^4 \log(p_n)\right)$$

The smoothing is computed by formula $p_n = \frac{1}{2^k}$ for each precision score whose matching n -gram count is null in all references. Here, $k = 1$ for the first n -gram precision score for which the n -gram count is null and is incremented by 1 for each of the following n .

Considering the task of image description generation, it is hard to believe that even the majority of possible components and actions in the scene can be captured by at least one of the references. Therefore, the scene descriptions, fluent and relevant to the scene, which will have no similar references, will be generated and will result with low BLUE score.

Because of the problems above, BLEU metric is not the best measure in evaluating image descriptions, however it is a widely used approach, and in order to be able to compare the results with other related work we decided to evaluate our models using (smoothed) BLEU score.

2.3.2 Meteor

METEOR (Metric for Evaluation of Translation with Explicit ORdering) is another metric for evaluating machine translation tasks introduced by Denkowski and Lavie (2014). As BLEU score has limited ability to capture the variability of the language used to describe images, Meteor was designed to fix this problem and still correlate well with human judgements. The main feature in Meteor is the improvement in matching two words. BLEU only considers exact matching whereas Meteor includes stemming, synonymy and paraphrase matching.

The Meteor score is calculated in the similar way as BLEU, however instead of scoring a candidate with respect to all reference sentences, it considers one reference sentence at a time and only keeps the score for the highest scoring reference. Meteor performs on unigram precision P and recall R scores where unigram precision is calculated as in BLEU. Unigram recall is a ratio between the unigram counts in the candidate sentence that are also found in the reference and total number of unigrams in the reference sentence. While matching words between candidate and reference sentences for recall and precision calculation stems, synonyms and paraphrases are also considered for matching.

Brevity penalty used in BLEU evaluation is exchanged by the fragmentation penalty score as:

$$\text{Pen} = \gamma \cdot \left(\frac{ch}{m} \right)^\beta$$

where ch is a number of sequences of matches that are adjacent and identically ordered in candidate and reference sentences and m is a total number of matched unigrams.

Finally, the Meteor score is calculated as:

$$\text{Score} = (1 - \text{Pen}) \cdot F_{\text{mean}}$$

where F_{mean} is a harmonic mean of P and R :

$$F_{\text{mean}} = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R}$$

The parameters α , β and γ are adjusted to maximize correlation with human judgements.

Meteor fits well in the task of image caption generation since it concentrates on semantic similarity rather than trying to find the exact matches between candidate and reference sentences. Meteor metric should correlate better with human judgements than BLEU because of its ability to capture variability of the sentence.

2.3.3 Qualitative Evaluation

As already described in this section, humans do not agree well in the way they describe the same image, however both automatic evaluation metrics use the set of human written captions as a gold-standard reference to score trained model. The limits of reference sentences expressibility also set a bound for new generated captions to capture the concept that is already captured by one of the references. Therefore, to overcome this problem qualitative evaluation has to be done manually by exploring the nature of captions generated by the models built.

Chapter 3

Experimental Set-up

This chapter presents how the methodology explained in Chapter 2 is implemented.

The detailed explanation of model building pipeline is presented in Section 3.1.

Section 3.2 details how the features, both textual and visual, are selected to be successfully used by the multimodal neural language models described in Section 2.2.

3.1 Building the model

This section explains the steps taken to build models introduced in Section 2.2. All three neural language models require a set of parameters to be specified prior to training. The selection of word and visual feature representation is explained in Section 3.2 whilst this section concentrates on model training parameters.

As described in Section 2.1, the Abstract Scenes Dataset includes the set of 3 to 9 short, one sentence descriptions for each scene. 20% of these scenes are left for testing final models in Section 4.1. Another 20% of scenes are left as a validation set for tuning model parameters and selecting best features in Section 3.2. To keep the sentence representation simple, each description in the remaining training set is considered as a separate instance. Therefore, multiple training captions refer to the same training image.

As a starting point, training parameters that were found as optimal by Kiros, Salakhutdinov, and Zemel (2014) were selected. They were then tuned on the validation set using the set of features found as optimal in Section 3.2.

The following parameters were selected for the final evaluation in Section 4.1. The initial model learning rates of 0.2 (LBL and MLBL-B) and 0.02 (MLBL-F) are selected which are exponentially decreased after each training iteration by a factor of 0.998. All weights (except word representation vectors which selection is described in Section 3.2.1) were randomly initialized from a zero mean Gaussian with a standard deviation of 0.01. As presented in Figure 3.1, the length of the training sentences does not vary a lot (remember that captions are usually of one sentence only and each of them is

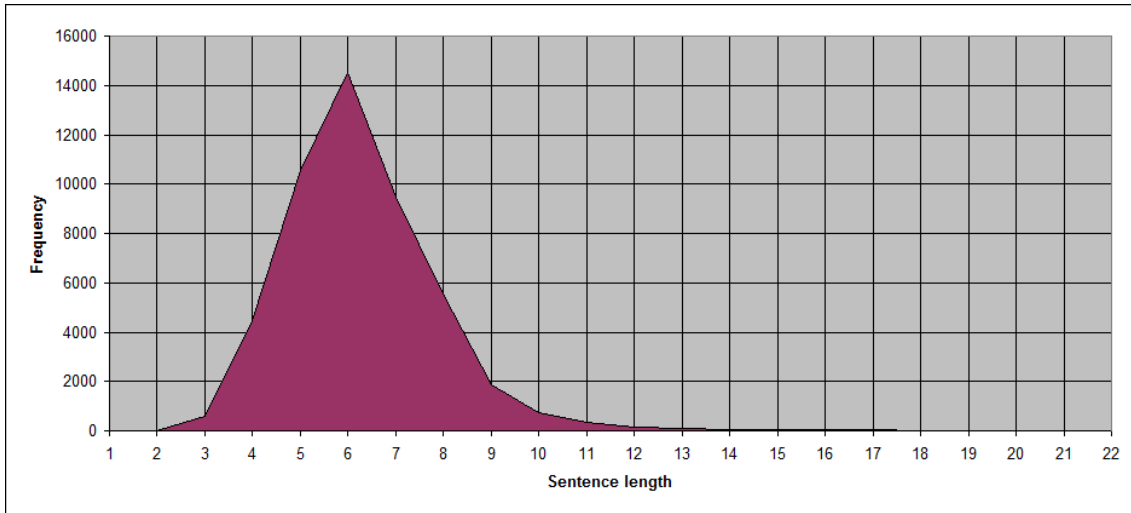


Figure 3.1: The frequency of the training sentences length

considered as a different training instance) and thus, all possible values of context size were tested. The optimal context size of 5 was found showing that models can be best trained by keeping the knowledge of all previously generated words starting from the beginning of the sentence but no more than that. Specifically for *MLBL-F* model, the number of matrix parametrization factors f parameter is additionally optimized. The tests were done for factors in range of 30 to 200 and it was found that model trained with TODO factors perform best.

Finally, the training data was cleaned by fixing spelling mistakes (see Appendix A for details) and all final models were then learned using backward propagation as described in Section 2.2.1.2. Captions were generated by applying the procedure presented in Section 2.2.3. All models were evaluated by collecting a set of generated captions, scoring it against the gold standard corpus using (smoothed) BLEU and Meteor metrics introduced in sections 2.3.1 and 2.3.2 and performing qualitative analysis in Chapter 4.

3.2 Feature selection

This section introduces how the features required for multimodal neural language models to be build were selected.

First, the selection of word features is explained in Section 3.2.1 and then, two visual feature sources are presented in Section 3.2.2.

3.2.1 Word features

The descriptive sentence of the image can be seen as a sequence of words where each word has some semantic meaning, sometimes dependent on context words. Words,

Word vector representations	(smoothed) BLEU	Meteor
Randomly initialized	17.75	21.37
25 dim embeddings	18.67	23.72
50 dim embeddings	21.07	23.23
100 dim embeddings	17.75	21.37
200 dim embeddings	17.41	23.05

Table 3.1: (smoothed) BLEU and Meteor scores for LBL model trained with randomly initialized word representations and Collobert & Weston embeddings of different dimensions

as a default string representation, do not capture semantically meaningful information that could be easily understood by computers, for instance it is not possible to compute whether two different words are similar by only knowing two word string representations. Therefore, smarter approach should be considered - words represented as vectors of features.

All three models (LBL, MLBL-B and MLBL-F) introduced in Section 2.2 are trained on word representation vectors which are not present in Abstract Scenes Dataset. Neural Language Models are able to update the word representation vectors by themselves after training each given instance (applying back propagation), however the initial word vectors must be still supplied.

The simplest approach to initialize word representation vectors would be to set them randomly (with a default representation vector size) and hope that the models will learn them. However, in order to improve the prediction accuracy (Turian et al., 2010b), pre-trained word embeddings can be used to initialize word vector representations. As Turian et al. (2010b) show, embeddings learned using neural language approach presented by Collobert and Weston (2008) can be adapted well and should be able to produce better results. Turian et al. (2010b) learned the embeddings of various dimensions (25, 50, 100 and 200 dimensions) that are publicly available online (Turian et al.).

To select one set of embeddings and evaluate the effectiveness of using pre-trained word embeddings, we trained a LBL model with each set of embeddings as well as randomly initialized word representations. Since better embeddings can only increase the value of language model and it has no additional effect on presented modality (an image) the results found for LBL model will be applicable to MLBL-B and MLBL-F models as well. Each model was evaluated on a validation set by collecting generated caption for each scene and measuring both (smoothed) BLEU and Meteor scores for a full set of proposed captions.

Table 3.1 presents evaluation scores for each trained LBL model. As hypothesised, the model with no pre-trained embeddings is scored the lowest¹ by both BLEU and Meteor metrics while the highest BLEU and Meteor scores are given to embeddings of dimensions 50 and 25 respectively. As a result, 50 dimensional pre-trained word

¹Interestingly, embeddings of 100 dimensions are not better than random initialization.

embeddings were chosen for all the remaining experiments throughout this project to optimally increase both (smoothed) BLEU and Meteor scores.

3.2.2 Visual features

This section describes visual feature sets that were used to train and compare multi-modal neural language models.

Section 3.2.2.1 presents a baseline feature set acquired by training the convolutional neural network. These features are learned directly from the images as a standard approach.

Section 3.2.2.2 shows a different approach to collect the set of visual features. It uses features from Abstract Scenes Dataset as described in Section 2.1 to find the optimal set of features for models training.

3.2.2.1 Convolutional features

Kiros, Salakhutdinov, and Zemel (2014) suggest learning image features using convolutional networks based pipeline described by Coates and Ng (2011) and use the outputs either as an additive bias (MLBL-B model) or for gating (MLBL-F only).

Convolutional networks is a trending approach in learning image features. Unlike standard multilayer neural networks that fully connect all the input units to all the hidden units, convolutional networks use convolution and pooling methods to have a neural network that is easier to train, have fewer parameters than fully connected network with the same number of hidden units and results with translation invariant features.

In general, convolutional networks design image's property of being immobile, meaning that various image statistics for all parts of the image are the same. As a result, convolutional networks learn features for a small part of the image at a time and applies them for all other parts (in other words, *convolve* them with a larger image) instead of learning the same features for each part of the image separately. After obtaining the convolved features pooling method is used to reduce the number of resulting features to decrease the threat of over fitting. It is achieved by dividing image into regions again and finding the max value (max-pooling) of a particular feature over a region of the image. More detailed explanation of convolutional networks and the parameters used is given by Kiros, Salakhutdinov, and Zemel (2014).

ConvNet software² were used to extract 4096 features from each scene. Models built with Abstract Scenes features described in the following section were compared with models using convolutional network's (abbreviated just ConvNet) features which indicates an absolute baseline. Automatic evaluation results and human based analysis of the models trained using ConvNet features are presented in Chapter 4.

²The code was downloaded from <https://github.com/TorontoDeepLearning/convnet>

Feature set	Feature count	BLEU	Meteor
Occurrence	58	41.57	32.95
Absolute depth	174	40.45	32.83
Relative location	1508	40.02	32.74

(a) Top 3 MBL-B models with singleton features.

Feature set	Feature count	BLEU	Meteor
Occurrence	58	41.76	33.26
Co-occurrence	377	38.97	31.60
Absolute depth	174	38.53	31.93

(b) Top 3 MBL-F models with singleton features.

Table 3.2: (smoothed) BLEU and Meteor scores (%) for 3 best MBL-B and MBL-F models trained with different singleton feature sets.

Features	MBL-B		MBL-F	
	BLEU	Meteor	BLEU	Meteor
All but rel. loc. with dir.	39.56	33.10	39.58	33.05
Al	38.49	32.59	39.32	33.23
All but relative location	38.78	32.42	38.05	32.03

Table 3.3: (smoothed) BLUE and Meteor scores (%) for MBL-B and MBL-F models trained with all features concatenated together as well as with all features excluding one of the relative location or relative location with direction feature sets. Scores in bold highlight models that perform worse than the model trained with all available features. All features excluding relative location with direction feature set performs better than with this set included for training stating the fact that relative location with direction feature set makes a negative effect in a value of generated captions.

3.2.2.2 Abstract Scenes features

The Abstract Scenes Dataset is the main source of scene features that are used for training multimodal neural language models throughout this project. As described in Section 2.1, image features in Abstract Scenes Dataset are split into different sets that each measures a single property of the scene and can be classified as discrete or continuous.

To select the combination of features that maximizes the accuracy of captions generated, number of various experiments were done. Since there are 10 different sets of features, combining and experimenting with each possible combination of feature sets is inefficient and thus, the decision was made to only do the following experiments (train MBL-B and MBL-F models):

All features. At first, all features were concatenated and used as a single set of features to find how well both models perform. Table 3.3 presents the results for MBL-B and MBL-F models trained with all 10 feature sets concatenated together.

Singleton features. 10 experiments were done to build both models with each feature set alone to get an intuition about what features are most semantically meaningful. The results for best 3 MBL-B and MBL-F models with singleton features are shown in Table 3.2. Both models trained with occurrence features only perform the best and surprisingly better than the model built with all possible Abstract Scenes features. Absolute depth feature set is amongst best 3 for both MBL-B and MBL-F models as well, although the score for MBL-F model is significantly lower than the one achieved with occurrence features.

All but one feature set. Another batch of experiments were done to build both models with one single set of features removed from full consideration mainly to check if all feature sets add some positive value into the final score. The most important observation made at this stage - not all sets of features add some positive value, in fact, the feature set containing relative location information that takes into account the direction an object is facing makes a negative effect and without this set of features both MBL-B (BLEU and Meteor scores increases by 1.07% and 0.51% respectively) and MBL-F (BLEU drops by 0.18% while Meteor increases by 0.26%) models perform better. The results are shown in Table 3.3. On the other hand, if relative location feature set is absent the scores are one of the lowest amongst all the models, thus showing that relative location is a valuable feature while training a model, however it is a facing direction that have negative impact on the results³. Consequently, the relative location with direction feature set was ignored in the remaining experiments. All other sets of features add some positive value since both metrics reduced for both MBL-B and MBL-F models.

Combinations of feature classes. Another set of experiments were done separating features into 3 classes: occurrence (includes object occurrence, two objects co-occurrence and the occurrence of person's pose or facial expression), position (both relative and absolute locations⁴ as well as both relative and absolute depths) and appearance (distance between the object and person's hand or head) features and building MBL-B and MBL-F models with each combination of these classes. The results are presented in Table 3.4. The feature set containing all occurrence and appearance class features performs better than any other feature class combination. In fact, next best model (which is the model built by using all feature class together) is scored 2.39% and 1.04% lower by BLEU and Meteor respectively, thus showing that to maximize generated captions quality in terms of automated measurements continuous position features are unnecessary and even harmful. On the other hand, position features in the combination with occurrence class maximizes the performance for MBL-F model.

Final tests. So far we found that MBL-B and MBL-F models perform best if occurrence+appearance and occurrence+position classes are used for training respectively. By comparing all observations so far, it can be seen that the models trained with features from a whole occurrence class⁵ are scored lower than models trained with fea-

³The models built with a singleton relative location features also performed better than the models trained with a singleton relative location with direction features.

⁴Note that relative location with direction feature set was not included into position or any other class since the presence of these features reduce the overall performance.

⁵Occurrence class represents occurrence, co-occurrence and the occurrence of person's pose or facial

Feature class	Feature count	MLBL-B		MLBL-F	
		BLEU	Meteor	BLEU	Meteor
Occurrence	459	39.53	32.72	40.86	32.63
Position	3045	36.18	31.42	37.37	32.00
Appearance	232	39.76	32.44	31.54	29.06
Occurrence + Position	3504	38.99	32.70	41.43	33.37
Occurrence + Appearance	691	41.95	34.14	39.12	32.33
Position + Appearance	3277	36.16	31.59	39.08	32.79
Occurrence + Position + Appearance	3736	39.56	33.10	39.58	33.05

Table 3.4: (smoothed) BLEU and Meteor scores (%) for MLBL-B and MLBL-F models trained with different feature class combinations. Clear winners are shown in bold. For MLBL-B model Occurrence+Appearance feature set outperforms all the remaining sets while Occurrence+Position feature combination works best for MLBL-F model with Occurrence feature set alone performing nearly as good.

tures from occurrence set only⁶. Since both appearance features combined together are scored higher than with each of them considered separately⁷, it is likely that the current best models can be further improved by training new MLBL-B and MLBL-F models with occurrence class features exchanged by occurrence set ones. It was found that both MLBL-B and MLBL-F models trained with occurrence set’s and appearance class features outperform all previously trained models and thus these features were selected for further consideration and comparison against other models in Chapter 4. The results for these two models are presented in Table 4.1.

All models were trained using default MLBL-B and MLBL-F training parameters as described by Kiros, Salakhutdinov, and Zemel (2014). This is, because both model parameters are independent in measuring which subset of features results with the highest (smoothed) BLEU and Meteor scores as long as the parameters are the same for all experiments. All models were evaluated on the validation set by generating a single caption for each instance and scoring them against gold-standard human references.

By applying different experiments described above, most influential features in terms of semantically meaningful information were found. First, we found that object occurrence features are the most semantically helpful while object’s facing direction is even harmful while learning both multimodal neural language models capable of generating meaningful descriptions. We have then explored which feature class combination can perform the best and it was found that occurrence class features are the most semantically meaningful while the addition of appearance or position features (MLBL-B and MLBL-F models respectively) improved the overall results. Finally, we

expression.

⁶BLEU and Meteor scores are higher by 2.04% and 0.23% respectively for MLBL-B and by 0.9% and 0.63% for MLBL-F models.

⁷Distance to head features acquire 29.52% (BLEU) and 28.02% (Meteor) scores for MLBL-B, 19.16% (BLEU) and 23.58% (Meteor) for MLBL-F models while distance to hand features are scored 35.07% (BLEU) and 29.75% (Meteor) for MLBL-B, 31.28% (BLEU) and 27.64% (Meteor) for MLBL-F models. Models trained with both feature sets separately perform worse than by using both appearance sets together as scored in Table 3.4.

discarded co-occurrence and occurrence of person's pose or facial expression features to further improve our both MLBL-B and MLBL-F models resulting with the set of 290 features including object's occurrence and object's distance to both hand and head that were found to be most helpful in the task. The fact that models with just 290 features can outperform the performance of the models using more or all sets of features concatenated together already shows a promising future in further real-photo description generation research⁸.

⁸Considering real photo-based images for which such feature sets cannot be extracted as simply as for Abstract Scenes Dataset and even if successfully extracted it is noisy due the problems with computer vision techniques, the foundation of models that can generate competitive descriptions by using just the most important features is preferred.

Chapter 4

Results

4.1 Automatic evaluation

In this section, the overview of the models, introduced in Section 2.2, performance in terms of automatic evaluation is given. Models trained with the best Abstract Scenes feature combination found in Section 3.2 are scored with (smoothed) BLEU and Meteor metrics described in Section 2.3 and compared against baseline models built with features extracted using computer vision techniques as outlined in Section 3.2.2.1. The performance of LBL model that generates captions without using a knowledge of image features is also included for reference.

As hypothesised in Section 2.3, both evaluation metrics (and especially BLEU) are not perfect for scoring image captions, because they are bounded by few human annotated descriptions as a reference for each scene. An example scene and a set of human descriptions are shown in Figure 4.1. Even in the scene with just a few objects and few actions present, humans still tend to describe it differently. Only two sentences involving Mike kicking the ball are very similar, others describe other object in the scene: dog, apple tree or lightning. Therefore, 6 reference sentences per image is far not enough to cover all the *meaning* presented by the image. To better understand the scale of the problem, the human performance in the same task of describing clip art images is measured.

Human performance is calculated as a human agreement on the Abstract Scenes Dataset. Specifically, each human description were scored in the same way as captions generated by trained models using both (smoothed) BLEU and Meteor metrics. That is, the relevance and accuracy of human description were measured against all the remaining references for the scene. For fair comparison, only the descriptions for test scenes¹ were considered in approximating human annotators agreement for the Abstract Scenes Dataset.

Table 4.1 summarizes the results of how well humans describe images compared with other human annotated captions. Human performance in terms of automatic evalua-

¹20% of all images that were chosen to test the performance of built models throughout the project.

Model	Feature count	BLEU score	Meteor score
LBL	0	21.07	23.23
MLBL-B ConvNet	4096	27.74	26.61
MLBL-F ConvNet	4096	28.46	26.80
MLBL-B occHandHead	290	44.07	34.14
MLBL-F occHandHead	290	43.24	34.92
Human Agreement	-	21.61	26.71

Table 4.1: Comparison of automatic evaluation scores for baseline language model and multimodal neural language models trained with ConvNet and occHandHead feature sets, as measured by (smoothed) BLEU and Meteor metrics. Here, ConvNet corresponds to the features extracted using computer vision techniques and occHandHead is a subset of the Abstract Scenes features that were found to perform the best. Human agreement on the Abstract Scenes Dataset is included as a reference to enable better interpretation of the results.

tion is relatively low. Especially considering that all these human captions are fluent, relevant and basically perfect descriptions the results prove the fact that automatic evaluation is not yet capable to correlate well with human judgements in the task of caption generation. This becomes a real problem for scoring new generated captions with respect to "gold-standard"² descriptions and Table 4.1 illustrates the problem well. Perfect captions, in the example shown, are generated by two models only, namely MLBL-B ConvNet and MLBL-B occHandHead³, however because of the limiting reference set with not a single mention of the hat boy is wearing both of these two perfect captions will be poorly scored. On the other hand, caption *Jenny is kicking a soccer ball* generated by MLBL-F ConvNet model will be scored the best since it is only one word, *Jenny*, that is missing in one of the reference *Mike is kicking a soccer ball*.

Two examples, the humans performance score and automatic evaluation's disability to correlate with human judgements, reveal the need of human based performance analysis to make sure the models automatically scored the best are indeed the best. The performance analysis is presented in the next section - Section 4.2.

Continuing with automatic evaluation, Table 4.1 also presents the results for three baselines - LBL and two multimodal models trained using convolutional features - as well as two best models built with a subset of Abstract Scenes features - occHandHead. Only one model, namely LBL baseline, performs worse than human agreement and it is as expected since this model always produces the most likely sentence independently from the scene image supplied. Apparently, the sentence *Jenny is wearing a hat* is found by the model to be the best fit amongst all training sentences and thus, is always outputted as an answer to any test image. Note, however, that even if the LBL model is scored lower than overall human agreement by both automatic metrics, the latter is not measured as significantly better. For instance, BLEU and Meteor scores for human

²Human created captions are used as a gold-standard set for scoring descriptions generated automatically, however they are not representative enough to be trusted 100%.

³occHandHead represents the subset of Abstract Scenes features including object's occurrence and object's distance to kids hand and head features.



Figure 4.1: The complete set of human annotated descriptions as well as captions generated by each of the final models for the sample Abstract Scenes image.

Human:

- (1) *Mike is kicking a soccer ball.*
- (2) *There are apples on the tree.*
- (3) *Lightning is coming from the cloud.*
- (4) *Lightning is striking.*
- (5) *Mike is kicking the soccer ball.*
- (6) *The dog is near the apple tree.*

LBL:

Jenny is wearing a hat.

MLBL-B ConvNet:

Mike is wearing a hat.

MLBL-F ConvNet:

Jenny is kicking a soccer ball.

MLBL-B occHandHead:

Mike is wearing a winter hat.

MLBL-F occHandHead:

Mike is wearing a blue cap.

performance are only 0.54% and 3.48% respectively higher than for LBL model. Such finding once again confirms that humans tend to describe the same image completely different and the chance of two annotators capturing the same aspect in the Abstract Scenes image is just slightly higher than the probability that the annotator describing specific image is using the concept covered by the most probable caption amongst all Abstract Scenes.

The performance of two baseline multimodal neural language models trained with convolutional features are scored as good as human agreement as measured by Meteor and 7-8% higher as measured by BLEU. The observed scores does not necessarily tell that descriptions retrieved by any of these models are better than the ones created by humans, however it can be formally interpreted that if x is the number of overlapping phrases in the candidate caption generated by MLBL-B or MLBL-F model using convolutional features that are also included in any reference then it is likely that on average any of these references alone will have lower number of phrases $y_i, i \in [1, \dots, n]$ that are also found in any remaining references R_{-i} :

$$\frac{1}{n} \sum_i y_i < x, i \in [1, \dots, n]$$

The good example of this property would be a sentence that combines multiple gold-standard concepts from different references (for example, caption *Lightning is striking Mike* for the scene in Figure 4.1). Therefore, models trained using convolutional features are not better than human annotated descriptions, however they tend to combine human references to maximize both automatic evaluation scores.

Finally, both MLBL-B and MLBL-F models that use occHandHead features from Abstract Scenes Dataset significantly outperform both baselines and human performance. As scored by BLEU, our trained Abstract Scenes models perform more than two times better than humans and around 30% better in terms of Meteor. Once again, it does not mean that these captions are **actually** better or more representative than human descriptions. As mentioned in previous paragraph, the model learned how to combine all concepts captured in training sentences and depending on the scene features, it tries to generate a description that includes as many of these concepts as possible or at least the most important ones⁴ so that both BLEU and Meteor would score it well against all references.

To sum up the results of automatic evaluation, it was showed that MLBL-B and MLBL-F models trained with the occHandHead feature set found in Section 3.2.2.2 significantly outperform both baseline and multimodal models built with ConvNet features as measured by automatic evaluation⁵. Interestingly, differences between MLBL-B and MLBL-F model, trained on the same set of features, scores are minor showing that it is a selection of visual features (and not the specific model) that matter the most when generating relevant image descriptions. Unfortunately, automatic metrics do not score well

⁴The importance learned by multimodal language models represents how often the concept in the scene is actually captured in the corresponding human annotation.

⁵The difference in models', trained with ConvNet and occHanHead features, scores confirm how noisy and unreliable computer vision is.

Abstract Scenes gold-standard references that are considered being of lower value than the actual generated captions by any of the multimodal neural language models. The problem could be solved by having more human annotated references to increase a captured variability of concepts in the scene. However, since the task of such data collection is hard and involves real human work, manual performance analysis of the models is done in the next section.

4.2 Human based performance analysis

In this section the more in depth analysis of the models performance that were scored automatically and quickly reviewed in previous section is done. The aim of the analysis is to confirm that the automatic scores given for the models described correlates well with human judgements. Note that even if human agreement score is relatively low and shows the huge pitfall in automatic evaluation metrics it might still be the case that models scored higher are in general better than the ones scored lower (especially in our case, where the difference in scores between models trained using Abstract Scenes and convolutional features are relatively high).

The following sections analyse multimodal neural language models trained with different set of features in detail with the final observations presented in Section 4.2.3. Note, since the LBL model is an absolute baseline that does not consider image features and always describe an image with the same caption, namely *Jenny is wearing a hat.*, there will be no more further analysis done for this model since it clearly performs worse⁶ than any other multimodal neural language model.

4.2.1 ConvNet features performance

In this section descriptions generated by both MBL-B and MBL-F models trained using the set of 4096 features extracted from the image using the machine vision techniques presented in Section 3.2.2.1 are discussed and compared with the descriptions created by the same models but using the best⁷ subset of the Abstract Scenes features, namely occHanHead.

By reviewing the descriptions manually it was found that there are scenes for which both sets of features help create good captions, however for some images neither model could output a reasonable description. To illustrate these findings, the set of scenes and their predicted descriptions representing the statements mentioned were sampled from the test set and are presented in Figure 4.3 and Figure 4.4. Note that all captions, independent of their relevance to the scene, are fluent and grammatically correct. This holds for all captions generated by any of the language models and shows that neural networks are capable to learn and use English language fluently.

⁶As automatically evaluated in Section 4.1

⁷As scored by both BLEU and Meteor.



Figure 4.2: Sample scene presenting the limitation of Abstract Scenes occurrence and distance to hand/head feature set. The small distance between two children is not always enough for the model to make a right decision of who is a subject of an action.

MLBL-B ConvNet: *Mike is wearing a blue shirt.*

MLBL-F ConvNet: *There is an airplane flying in the sky.*

MLBL-B occHandHead: *Mike and Jenny are playing with a beach ball.*

MLBL-F occHandHead: ***Mike is flying a kite.***

Sample human description: *Jenny is flying a kite.*

Although the previous examples showed that both ConvNet and Abstract Scenes features can be equally good/bad in training neural language models of high performance, however in general ConvNet features rarely perform better than the best Abstract Scenes ones. In fact, this happens not because ConvNet performs better for some particular cases but due to limitations of the best Abstract Scenes features set. The example scene is shown in Figure 4.2. These limitations are further described in Section 4.2.2.

To better understand what is the main problem in the descriptions generated by models using ConvNet features, Figure 4.5 presents example scenes for which Abstract Scenes features based models generate more accurate descriptions. In general, MLBL-B and MLBL-F models trained using ConvNet features are both unreliable and unpredictable. The main problem of these models is a disability to understand what objects are in the scene and thus, most often outputs an object or relation that is not present in the image whereas models using Abstract Scenes occurrence features are more stable in this task. The scenes with one or both missing kids are one of the hardest for ConvNet models to handle and they usually fail to not mention the missing kid's name in the description.

By analysing and comparing descriptions generated by the multimodal neural language

models trained using convolutional and Abstract Scenes features, the limitations of machine vision techniques were exposed. Learned image features are not reliable and cannot get anywhere close to the Abstract Scenes features. Having the latter, stable models can be learned that are capable of at least recognizing the basic relations between the objects that are surely⁸ present in the image.

The findings of this section confirm the relative automatic evaluation scores given to the models described are correct. Models trained with ConvNet features are indeed less likely to produce relevant to the scene descriptions whereas captions generated by models trained with Abstract Scenes `occHandHead` features are known to be of higher quality almost every time. The differences in automatic scores between `MLBL-B` and `MLBL-F` models for the same set of features are also consistent with the findings - neither `MLBL-B` nor `MLBL-F` perform better than the other.

4.2.2 Abstract Scenes features performance

After having serious concerns whether the automatic evaluation, in particular (smoothed) BLEU and Meteor metrics, is consistent with human judgements it is not yet clear that the models trained using `occHandHead` features from Abstract Scenes Dataset are better than any other feature subset in the task of meaningful captions generation. Therefore, one different subset of Abstract Scenes features was selected to be manually compared with the subset used so far. As a result, a set of all occurrence and position class⁹ (later called as `OccPos`) features was selected for further comparison since the set consists a more various features and the models built with these features were also scored high. For fair comparison, both multimodal language models using `OccPos` features were trained with cleaned training sentences (see Appendix A).

Prior to comparing two Abstract Scenes feature sets, the main problems with `occHandHead` set are described. In general, the descriptions are fluent, grammatically correct and most of the time relevant to the scene. However, Figure 4.4 presents the example scenes for which neither `MLBL-B` nor `MLBL-F` manage to generate accurate captions. Note that even if the captions are irrelevant, all subjects and objects that are mentioned in the description are present in the scene. Often even the action that is described in the sentence is happening in the scene, but involving different subject and/or an object. As an example, for the first image in Figure 4.4 `MLBL-B` and `MLBL-F` models generate the following captions: *The soccer ball is in the sandbox.* and *Mike is sitting in the sandbox.* None of them is correct, however both subjects, *soccer ball* and *Mike*, an object *sandbox* and both action concepts, *is in the sandbox* and *is sitting in the sandbox*, are presented by the image.

The problem can be explained by considering the features the models are using. The only features that are known by the models are discrete object's occurrence and continuous distances from all the objects in the scene to the children's hand or head. The presence of occurrence features explains why both models are able to spot the right

⁸If occurrence features are present, the models are significantly helped in recognizing the presence of all the objects in the scene.

⁹Features of occurrence and position classes are explained in Section 3.2.2.2.

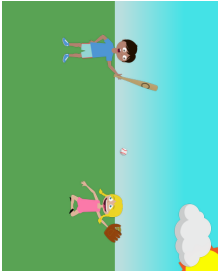

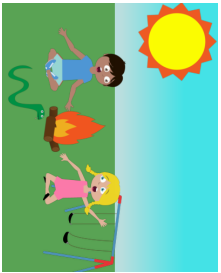
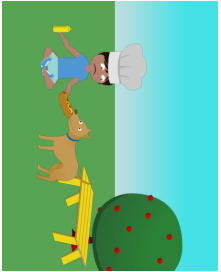
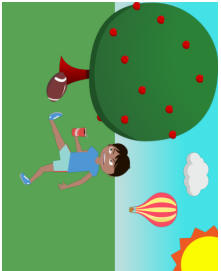
					
MLBL-B ConvNet	Mike is holding a bat.	Mike and Jenny are camping.	Mike is sitting next to the fire.	Mike is sitting on the grass.	Mike is standing by the tree.
MLBL-F ConvNet	Mike is holding a bat.	Jenny is sitting in front of the tent.	Mike is sad because he is sad.	Mike is wearing a chef hat.	Mike is near the apple tree.
MLBL-B ocCHandHead	Mike is holding a bat.	Jenny is sitting by the fire.	Mike and Jenny are sitting by the fire.	Mike is wearing a chef hat.	Mike is holding a cup.
MLBL-F ocCHandHead	Mike is holding a baseball bat.	Mike is wearing a baseball mitt.	Mike is sitting by the fire.	Mike is holding a hot dog.	Mike is holding a cup.
Human description	Mike hit the ball to Jenny.	Jenny fell down by the campfire crying.	Mike is next to the fire.	Mike is wearing a chef's hat.	Mike is kicking a football holding a soda.

Figure 4.3: Sample scenes that are well described by all multimodal neural language models.

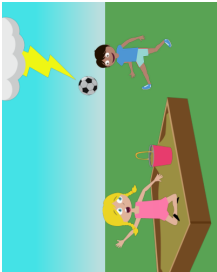

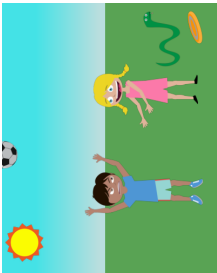
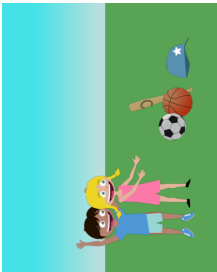

<div>MLBL-B ConvNet</div> <div>MLBL-F ConvNet</div> <div>MLBL-B occHandHead</div> <div>MLBL-F occHandHead</div> <div>Human description</div>					
	Mike is sitting in the sandbox.	Mike is wearing a hat.	Mike is wearing a hat.	The cat is sitting next to the cat.	Mike is wearing a pirate hat.
	Jenny is holding a tennis racket.	Mike is holding a kite.	Mike is holding a basketball.	Mike is wearing the hat.	Jenny and Mike are both wearing the hat.
	The soccer ball is in the sandbox.	Mike is flying a kite.	Mike is kicking the soccer ball.	Mike is wearing a baseball cap.	Jenny is waving to Mike.
	Mike is sitting in the sandbox.	Mike is flying a kite.	Jenny is running away from the snake.	Mike is wearing a baseball cap.	Jenny is holding a balloon.
	Jenny is in the sandbox.	Jenny is flying a kite.	The snake is moving away from Jenny.	Mike and Jenny have many different sports to play.	Lightning is coming from the cloud.

Figure 4.4: Sample scenes that are poorly described by all multimodal neural language models.

					
MLBL-B ConvNet	Mike and Jenny are playing baseball.	Mike is wearing a viking hat.	Mike is wearing a blue hat.	Mike is holding a shovel.	Jenny is very sad.
MLBL-F ConvNet	Mike is wearing sunglasses.	Mike is kicking the ball.	Jenny is holding a soccer ball.	Mike is in the sandbox.	The cat is by the table.
MLBL-B ocCHandHead	Mike is playing soccer.	Mike is wearing a hat.	Jenny is holding a tennis racket.	Jenny is wearing a hat.	The mustard is on the table.
MLBL-F ocCHandHead	Mike is playing with a soccer ball.	Mike is wearing a silly hat.	Jenny is holding a tennis racket.	Jenny is wearing a witch hat.	There is a cloud in the sky.
Human description	Mike is sad because he is alone.	Mike is standing next to the dog.	Jenny wants to play tennis.	Jenny is holding a baseball bat.	There's mustard on the table.

Figure 4.5: Sample scenes that are poorly described by multimodal neural language models that use ConvNet features.

objects and subjects in the scene whereas the presence of the distance to hand/head features creates another interesting effect. Note the general difference between the scenes for which good (Figure 4.3) and bad (Figure 4.4) descriptions are generated - the objects kids are either wearing or holding. For the scenes that were described well, the presented kids are wearing or holding a meaningful objects such as baseball glove, bat or hat, chef's hat, a cup or any other head accessory or item in hand whereas kids in poorly described scenes are less likely to wear or keep some particular item. Because of the small and limited `occHandHead` feature set, models prioritize describing what the kids are wearing or holding since these objects are the closest to either head or hand. As it turns out, the majority of the scenes in the Abstract Scenes Dataset include kids wearing a head accessory or holding some item in hand and thus, the model with `occHandHead` features can usually produce relevant captions. For the scenes that do not include kids with important¹⁰ head accessories or items in hand, models are trying to capture an action including other items around and usually fail since they do not have any other features such as position to use. As an example, even though Jenny is the one who holds a kite in Figure 4.2, MBL-F model captures that it is Mike who is holding it.

Interestingly, if none of the kids appear in the image, both MBL-B and MBL-F models can still produce relevant descriptions even though only the occurrence features are used¹¹. Descriptions for the example scenes with no kids in them are shown in Figure 4.6. Amazingly, just by using occurrence features both models were able to describe the facts such as *dog is looking at the cat*, *snake is under the tree* or *dog is wearing a pirate hat* showing that models successfully learned these facts as a likely to happen in Abstract Scenes world.

Finally, after looking closely at the problems and achievements of the models trained with `occHandHead` set of features, they are compared to models built with `OccPos` features. The comparison was done in two main steps. First, it was checked whether the poor descriptions generated using `occHandHead` features could have been fixed if `OccPos` features were used. Figure 4.7 presents descriptions generated by models trained with `OccPos` features for the scenes in Figure 4.4. Exactly half of these cases were decently described using `OccPos` set of features. Second comparison step involved scenes that were well described by `occHandHead` features to find out whether `OccPos` features can generate captions that are equally good. Figures 4.8 and 4.9 present how well both multimodal models perform on the scenes that are described well by the same models but using `occHandHead` features. Unfortunately, the bigger set of features, namely `OccPos`, including position information were not able to produce as relevant and fluent captions as `occHandHead` features. One of the main problems remains¹² incorrect object-action matching. However, it is no longer the case that at least all objects and actions mentioned in the sentence will be presented in the scene¹³.

¹⁰Important items are learned by the models themselves depending on how often the item that appears in the training scene is mentioned in associated gold-standard description.

¹¹If child is not in the image all distance to hand or head features are zeros and thus, do not provide any value.

¹²Note that the same problem was discovered for `occHandHead` features

¹³Although poor descriptions in Figure 4.8 do include all objects that are present in the image, that is not true for descriptions in Figure 4.9

Even a bigger problem can be seen in one of the generated sentences *Mike is wearing a ball.*. There is no need for a picture to understand what is wrong with the caption - language model is not that great as it was for models built with `occHandHead` features. Luckily, at least the scenes without kids were still most of the time well described by the new feature set as shown in Figure 4.6.

As compared, there is no one model that can produce better descriptions for all the scenes. Depending on the scene, models trained with `OccPos` feature set might produce better caption than models built using `occHandHead` features, however because of the `OccPos` set limitations in recognizing correct object and learning accurate language model, both `MLBL-B` and `MLBL-F` models trained using `occHandHead` features are favoured by us, humans and automatic evaluation metrics.

4.2.3 Discussion of the analysis

This section analysed the qualitative performance of multimodal neural language models, `MLBL-B` and `MLBL-F`, trained using different sets of features. It was shown that although the automatic evaluation is not reliable to quantify how good the models are in comparison with humans¹⁴, however the relative scores given to the models built are consistent with human judgements.

As a result, our found set of Abstract Scenes features that include only the object's occurrence and their distance to kid's hand or head is indeed the set of features that optimizes the performance of both multimodal neural language models. Captions generated by these two models are fluent, grammatically correct and most of the time relevant to the scene that makes them reliable and competitive. The biggest problem that arises from the features that are supplied to the models is limited creativity. Both models are most likely to describe a trivial aspect of the image provided by the objects kids are wearing or holding which is not always the most interesting aspect in the image.

Throughout the performance analysis, `MLBL-B` and `MLBL-F` models were never compared to each other. This is mainly, because both of these models trained on the same set of features perform equally well. This fact is supported by automatic evaluation scores and quality of the captions generated by both of these models. The only reason why `MLBL-B` might be favoured over `MLBL-F` model is slightly better creativity while describing images without children in it. As Figure 4.6 illustrates, `MLBL-F` model tends to capture and describe actions that are trivial given the object: *An airplane is flying in the sky* or *There is a thunderstorm.*. These captions are perfectly correct, however for humans might look boring and pointless.

¹⁴Humans agreement was scored lower than any of the model's, however it does not mean that multimodal neural language models can perform better than humans in the task of captions generation. It just shows that models tend to describe the combination of the most important and trivial aspects in the image, so that it likely agrees well with one of the gold-standard references.

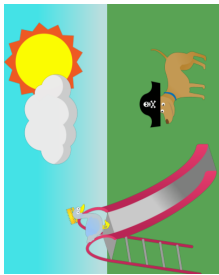

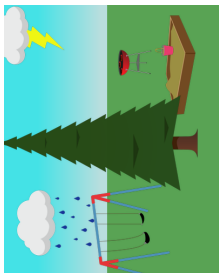
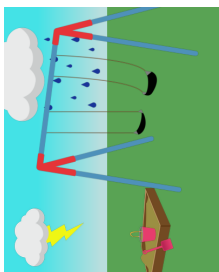
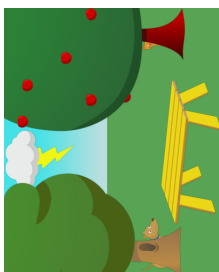
<div>MLBL-B occHandHead</div> <div>MLBL-F occHandHead</div> <div>MLBL-B OccPos</div> <div>MLBL-F OccPos</div> <div>Human description</div>					
	The dog is wearing a pirate hat.	The snake is under the tree.	The swings are empty.	The pail is next to the sandbox.	The dog is looking at the cat.
	The dog is standing by the slide.	An airplane is flying in the sky.	There is a thunderstorm.	The pail is in the sandbox.	The dog is standing by the tree.
	The sun is shining bright.	The snake is under the tree.	There is a cloud in the sky.	A ball is in the sandbox.	There is a cat in front of the tree.
	The dog is wearing a hat.	The owl is sitting on the ground.	There is a thunderstorm.	There is a shovel in the sandbox.	The dog is under the tree.
	The animals are wearing silly hats.	There are apples in the tree.	It is raining in the park.	It is raining on the swing.	The cat is behind the apple tree.

Figure 4.6: Sample scenes including no child described by multimodal neural language models using `occHandHead` and `OccPos` features extracted from Abstract Scenes Dataset.

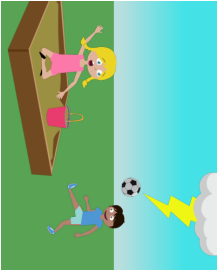
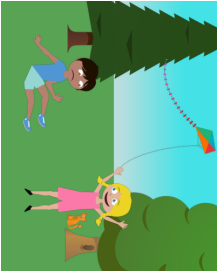
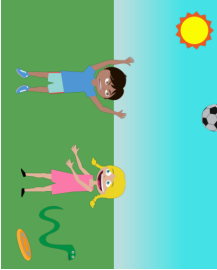
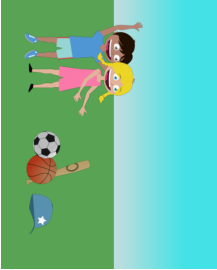
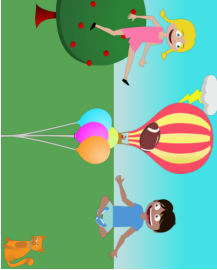
MLBL-B OccPos					
	Mike is kicking the soccer ball.	Mike is flying a kite.	Jenny is kicking the soccer ball.	Mike is happy to see the ball.	Mike is kicking the football to Jenny.
MLBL-F OccPos	Mike is kicking the soccer ball.	Mike is sitting under the tree.	Mike is kicking the frisbee.	Mike and Jenny are playing soccer.	Mike is kicking the football.

Figure 4.7: Sample scenes described by `MLBL-B` and `MLBL-F` models trained with `OccPos` features that were poorly described by the same models but using `occhHead` features as in Figure 4.4. Bold text represent sentences that are relevant to the scene and are of better quality than the ones created by `occhHead` features.



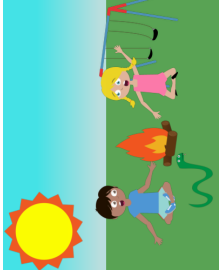
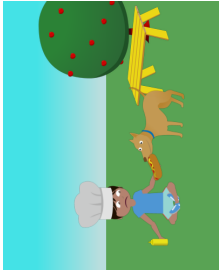
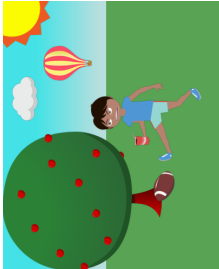
MLBL-B OccPos					
	Jenny is holding a baseball bat.	Jenny is wearing a baseball glove.	The snake is next to the fire.	Mike is wearing a chef hat.	Mike is mad.
MLBL-F OccPos	Mike and Jenny are playing baseball.	Mike is sad because he is crying.	Mike and Jenny are scared of the snake.	Mike is wearing a chef's hat.	Mike is kicking the ball.

Figure 4.8: Sample scenes described by MLBL-B and MLBL-F models trained with OccPos features that were well described by the same models but using occHandHead features as in Figure 4.3. Bold text represent sentences that are not relevant to the scene and are of worse quality than the ones created by occHandHead features.

MLBL-B OccPos				
	Mike is kicking the soccer ball.			
MLBL-F OccPos				
	Mike is wearing a blue hat.			
				
	Mike is wearing a ball.			
				
	Jenny is wearing a witch's hat.			
				
	The hot air balloon is flying over the park.			
				
	The grill is behind the table.			

Figure 4.9: Sample scenes described by `MLBL-B` and `MLBL-F` models trained with `OccPos` features that were well described by the same models but using `occhHandHead` features as in Figure 4.5. Bold text represent sentences that are not fluent or relevant to the scene and are of worse quality than the ones created by `occhHandHead` features.

Chapter 5

Conclusions

This report described the first part of the project done with the main goal to generate fluent and relevant to the visual scene descriptions by studying the Abstract Scenes Dataset introduced by Zitnick and Parikh (2013). Dataset containing simplistic clip art images with a set of high quality features and corresponding human annotated descriptions were crucial to investigate and successfully find the most semantically meaningful properties that can help describing clip art image.

In this chapter, the final comments are made to summarize the main contributions of the project in Section 5.1, discuss the main observations and possible further research in Section 5.2 as well as providing goals and plans for the second part of the MInf project in Section 5.3.

5.1 Summary of Contributions

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

First, the selection and adaptation of simple log-bilinear and two sophisticated multimodal neural language models were made as described in Section 2.2 to allow the simple usage of preprocessed Abstract Scenes Dataset including clip art images with associated feature sets and corresponding captions. Next, evaluation methods were carefully selected in Section 2.3 to make sure that the metrics used are reliable and correlated with human judgements in the complicated task of scoring generated image descriptions.

After considering the methodology to be used throughout the project the experimentation stage happened. During this stage, optimal word and visual features were found as explained in Section 3.2. As found in Section 3.2.2, object's occurrence and object's distance to the presented kid's hand and head features are the only features needed for semantically meaningful caption generation whereas the addition of any other features makes the models perform worse. The selection of optimal model parameters was described in Section 3.1 with the final improvement in the model's performance introduced in Appendix A. Here, the algorithm for fixing spelling mistakes in the Abstract

Scenes sentences is proposed that increased the overall performance by at least a small percentage (in terms of BLEU and Meteor) for each of the trained models.

Finally, baseline LBL and two multimodal neural language models, namely MBLB-B and MBLB-F were evaluated and compared for the different set of features. For comparison with the models trained using Abstract Scenes features, features extracted using machine vision techniques, described in Section 3.2.2.1, were used. Both MBLB-B and MBLB-F models trained using the optimal¹ subset of Abstract Scenes features significantly (in terms of BLEU and Meteor scores) outperformed baseline models that use learned features as stated in Section 4.1. To highlight the limitations of automatic evaluation, human performance was approximated as human annotators agreement on Abstract Scenes testing set. Shocking results revealed the need of human based performance analysis. The analysis was described in Section 4.2 which confirmed relative automatic evaluation's scores and demonstrated that multimodal neural language models trained with our proposed features selected from Abstract Scenes Dataset can reliably and consistently generate fluent and most importantly relevant to the scene descriptions.

5.2 Final remarks

By working with a concrete set of features where each of them describes a specific property such as occurrence, co-occurrence, position, we were able to experiment and find the set of image properties that are the most important for semantic meaning understanding. Even more importantly, we showed that more features do not necessarily produce higher quality captions, in fact, the opposite holds true in Abstract Scenes Dataset where the set of occurrence and distance to hand/head features only is the best feature combination to be used for semantic understanding.

We also realized that since Abstract Scenes Dataset is a broad collection of objects, actions, poses, expressions that cover a relatively big variety and complexity of semantic information found in real life situations, the results found should be applicable for the real photo-based images. With this in mind, if instead of extracting a huge set of features using various computer vision techniques only the features that are found to be semantically meaningful (as for instance occurrence and distance to hand/head) are extracted then it is likely that the quality of descriptions for photo-based images might increase. Note that the best subset of features we found only includes a small number of features. Therefore, extracting less features using computer vision techniques could substantially reduce the number of noise coming from other features that are not even meaningful in the task of image caption generation.

Another important observation made touches the way image descriptions are automatically evaluated. As it was shown in this report, current automatic evaluation metrics are not reliable in scoring image description. They do correlate with human judgements, however they are not capable of scoring the creativity of generated captions and thus, favour descriptions that are trivial observations in the image and do not represent the

¹Optimal subset corresponds to occurrence and distance to hand/head features.

meaning human would expect to be told. As such, new automatic evaluation metrics should be first investigated and discovered for this specific problem in order to at some point in the future have a system that could not only generate fluent and relevant descriptions, but also describe it creatively and capture the most crucial concept in the scene.

5.3 Future Work

Next year we are planning to continue working with Abstract Scenes Dataset as well as the multimodal neural language models and investigating the relations between scenes and their descriptions. However, unlike this year, the aim of the next year's project will be to generate clip art images that can visually represent the meaning of the description (or maybe even couple descriptions) provided. Note that this is a inverse problem that requires the same amount of initial data.

Therefore, we will be working with the same Abstract Scenes Dataset's scenes, their associated descriptions and given features. Since the most semantically important feature set was already found during this project, it will also be initially used to train models capable of generating scenes for given descriptions. However, it might be the case, that the semantic meaning for the inverse problem is best captured by a different set of features, thus, careful investigation of the features will have to be done again to prove or deny the goodness of occurrence and distance to hand/head feature set.

The advantage of Abstract Scenes Dataset that can ease the task is the provided scene renderer that can generate the image from the set of written information as shown in Figure 5.1. We will use that as a starting point with the aim to generate such information.

In terms of considerable models, we will try to use the inverted version of multimodal neural language models, however since it was not used for this kind of the problem before there will be a need of further investigation done to understand whether it is possible or other approaches has to be taken. To evaluate our models, new techniques will have to be considered since neither BLEU nor Meteor are created to measure the similarity between two images.



Clip art	Type index	Object index	X position	Y position	Depth	Flip
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

Figure 5.1: Sample scene and complete information needed to render the scene.

Appendix A

Cleaning the data

As claimed by Zitnick and Parikh (2013) the Abstract Scenes Dataset they have built, specifically scene captions, should be clean, written in clear English and hopefully without many spelling mistakes. However, after training few models and exploring the nature of captions generated it was clear that there are problems in the dataset. For instance, the following sentence was once generated: *The orange cat sees the nice day ion on the barbeque*. It has even two problems in it - misspelled word *barbeque* and the word *ion*. Even though the latter is a correct word, however it is hard to imagine why would anyone mention this word in the context of kids playing in the yard. Therefore, by checking the Abstract Scenes Dataset, the training sentence including word *ion* was found: *Mike has a big smile ion his fdace; he is happy*. It is now clear that the worker who was writing this description made a spelling mistake (in fact, even two).

After further investigation of the provided sentences quality it was decided to try cleaning the Abstract Scenes training sentences and hopefully increase the generated captions quality of the final `MLBL-B` and `MLBL-F` models. The cleaning done was very straightforward and only included spelling correction of the words that are surely misspelled¹.

The spelling correction procedure for each word in the training captions set is applied as follows: first, spelling of the word is checked using PyEnchant spell checking library introduced by Kelly. Next, if the word is found misspelled by Enchant library, the correct word version is found by using simple spelling corrector written by Norvig and as he claims it achieves 80-90% accuracy which is enough for us. The advantage of this model is that the gold-standard corpus² can be specified by the developer and thus, to correct misspelled words from the first half of the dataset the second half³

¹Therefore, the example word *ion* would not be fixed since it is a correct word, however it might be left as a future improvement.

²Gold-standard corpus is used to check how often similar words occur so that the decision of correct word version can be made. In this case *similar* words are defined as a set of words that has edit distance of 1 or 2 compared to misspelled word.

³In fact, Abstract Scenes Dataset includes all training captions as two separate corpus where each corpus has about a half (usually 3) descriptions for each scene. Moreover, multiple captions written by a single worker are always included in one of these two corpus, therefore worker specific spelling mistakes can be easily fixed.

was used as a gold-standard corpus and vice-versa. The word spelling was only fixed if at least 80% of the total similar words frequency in a specified corpus is covered by a single similar word which is then considered as a correct version of misspelled word. Example words that were affected by this model are shown in Table A.1. The majority of the misspelled words were successfully fixed, however the results are not ideal considering the incorrectly fixed words. Words that were poorly fixed are often a accidental concatenation of two words such as *infront*, *becauseit* or *standingby*. Since our spell corrector does not consider these cases they are most often changed by the longer word. However, because the truncated word is most often a preposition the resulting version of incorrect word still contains information about the main object, action or attribute that can be more accurately studied than by exploring a concatenation of words.

After applying the spelling corrector on the training sentences, final LBL, MLBL-B and MLBL-F models were re-trained for both set of features (extracted from convolutional networks and combinations of Abstract Scenes features) to explore the improvement in the resulting scores which has increased for all models by at least a small percentage in terms of (smoothed) BLEU and Meteor metrics.

Original word	Fixed word
frisbe	frisbee
Jennys	Jenny
sandwhich	sandwich
iwth	with
raquet	racquet
helment	helmet
weaing	wearing
Jenny	Jenny
behing	behind
wering	wearing
suprised	surprised
frizbee	frisbee
throughs	through
scarying	scaring
soccor	soccer
eatng	eating
smilling	smiling
rackett	racket
musturd	mustard
Jenyn	Jenny
playng	playing
becasue	because
Thier	Their
barbequing	barbecuing
becaues	because
picknic	picnic
beind	behind
runnining	running

(a) Correctly fixed words.

Original word	Fixed word
infront	front
sittingin	sittingon
onthe	ofthe
thru	the
ontop	onto
issittingin	issitting
iswearing	wearing
scaredof	scared
bythe	the
hitted	hitter
becauseit	because
isstanding	standing
standingby	standing

(b) Incorrectly fixed words.

Table A.1: Figure (a) shows example words that were correctly fixed while Figure (b) presents words that were incorrect and incorrectly fixed. Words that are misspelled were correctly fixed most of the time (except few cases such as *thru* and *hitted*), however our spelling fixer could not handle cases where two words are accidentally concatenated and thus, few mistakes shown in Figure (b) were made.

Bibliography

- Alex Krizhevsky et al, Geoffrey E Hinton. 2010. Factored 3-way restricted boltzmann machines for modeling natural images. In *International Conference on Artificial Intelligence and Statistics*, 621–628.
- Barnard, Kobus, Pinar Duygulu, and David A. Forsyth. 2001. Clustering Art. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, 434–441.
- Berg, Tamara L., Alexander C. Berg, and Jonathan Shih. 2010. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *Proceedings of the 11th European Conference on Computer Vision: Part I, ECCV'10*, 663–676. Berlin, Heidelberg: Springer-Verlag.
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Coates, Adam, and Andrew Y. Ng. 2011. The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization . In Lise Getoor and Tobias Scheffer, eds., *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 921–928. New York, NY, USA: ACM.
- Collobert, Ronan, and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, 160–167. New York, NY, USA: ACM.
- Denkowski, Michael, and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.
- Elliott, Desmond, and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 2 of *Short Papers*, 452–457. Baltimore, Maryland.
- Feng, Yansong, and Mirella Lapata. 2013. Automatic Caption Generation for News Images. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(4): 797–812.
- Grubinger, Michael, Paul Clough, Henning Muller, and Thomas Deselaers. 2006. The

- IAPR TC-12 benchmark: a new evaluation resource for visual information systems. In *International Workshop OntoImage*, 13–23.
- Heider, Fritz, and Marianne Simmel. 1944. An Experimental Study of Apparent Behavior. *The American Journal of Psychology* 57(2): 243–259.
- Kelly, Ryan. 2010. PyEnchant. <http://pythonhosted.org/pyenchant/>.
- Kiros, R., R. Salakhutdinov, and R. S. Zemel. 2014. Multimodal Neural Language Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 595–603.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, eds., *Advances in Neural Information Processing Systems* 25, 1097–1105. Curran Associates, Inc.
- Mcculloch, Warren, and Walter Pitts. 1943. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* 5: 127–147.
- Memisevic, R., and G. Hinton. 2007. Unsupervised Learning of Image Transformations. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 1–8.
- Mnih, Andriy, and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, 641–648. ACM.
- Norvig, Peter. 2010. How to Write a Spelling Corrector. <http://norvig.com/spell-correct.html>.
- Ordonez, Vicente, Girish Kulkarni, and Tamara L. Berg. 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Neural Information Processing Systems (NIPS)*.
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, 311–318. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Ramnath, Krishnan, Simon Baker, Lucy Vanderwende, Motaz El-Saban, Sudipta Sinha, Anitha Kannan, Noran Hassan, Michel Galley, Yi Yang, Deva Ramanan, Alessandro Bergamo, and Lorenzo Torresani. 2014. AutoCaption: Automatic Caption Generation for Personal Photos. *IEEE Winter Conference on Applications of Computer Vision*.
- Rashtchian, Cyrus, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. 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*, 139–147. Stroudsburg, PA, USA: Association for Computational Linguistics.

- Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6): 386–408.
- Turian, Joseph, Ratnovec Lev, and Bengio Yoshua. 2010a. MetaOptimize - Word representations for NLP. <http://metaoptimize.com/projects/wordreprs/>.
- Turian, Joseph, Lev Ratnovec, and Yoshua Bengio. 2010b. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, 384–394. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Zitnick, C. Lawrence, and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3009–3016. Portland, Oregon.
- Zitnick, C. Lawrence, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, 1681–1688. Sydney, Australia.