Evaluating Image2Speech

The evaluation of automatically generated phoneme captions for images

Justin van der Hout

Evaluating Image2Speech

The evaluation of automatically generated phoneme captions for images

by

Justin van der Hout

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday August 26, 2020 at 10:30 AM.

Student number:	4319982	
Project duration:	September 2, 2019 – August 26, 2020	
Thesis committee:	Dr. O. Scharenborg,	TU Delft, supervisor
	Dr. H. Hung,	TU Delft
	Prof. dr. ir. A. Bozzon,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Contents

Summary

Image2Speech is the relatively new task of generating a spoken description of an image. Similar to Automatic Image Captioning, it is a task focused on describing images, however it avoids the usage of textual resources. An Image2Speech system produces a sequences of phonemes instead of (written) words which makes the Image2Speech task applicable to languages which do not have a standardized writing system. This thesis presents an investigation into the evaluation of the Image2Speech task. The Image2Speech output is evaluated with human evaluators as well as multiple objective evaluation metrics. These metrics are often used in the field of Natural Language Processing, such as BLEU, METEOR, PER, etc. and can be used to give an indication of the semantic similarity between two sentences of words. Since humans are the end users of Image2Speech systems, the objective evaluation metrics are correlated with human evaluation in order to determine which metric can best evaluate an Image2Speech system with the end users in mind. For this, first an Image2Speech system was implemented which generates image captions consisting of phoneme sequences. This system outperformed the original Image2Speech system on the Flickr8k corpus, which is a dataset containing 8,000 images which each image also having five written and spoken captions. Subsequently, these phoneme captions were converted into sentences of words in order to be more easily interpretable for human evaluators. The captions were rated by human evaluators for their goodness of describing the image and correlated with the objective evaluation metrics. Although BLEU4 does not perfectly correlate with human ratings, it obtained the highest correlation among the investigated metrics, and is the best currently existing metric for automatically evaluating the Image2Speech task. Current metrics are limited by the fact that they assume their input to be words. A more appropriate metric for the Image2Speech task should assume its input to be parts of words, e.g. phonemes, instead.

1

Introduction

Automatic image captioning [32][46][12][9][13], the generation of descriptions for images, is a popular task that combines the fields of computer vision and natural language processing (NLP). Automatic image captioning can be helpful for human to robot interaction, early education, information retrieval (e.g. automatic tagging and easier searching), aiding the visually impaired, and more [2]. Image captioning systems typically use images with corresponding textual descriptions as training material, they are trained with image features as input and captions as the desired output. Unfortunately, such systems are only applicable to languages that have a conventional writing system (also known as a well-defined orthographic system). It is estimated that half of the 7,000 languages around the world however do not have such an orthography [10][44], for example Swiss-German.

In order to potentially reach any spoken language, regardless of whether it has an orthography, a new task has been proposed: Image2Speech [16], which takes an image as input and generates a caption as output. The main difference between Image2Speech and regular image captioning is that Image2Speech focuses on generating a spoken description without the use of textual descriptions. Rather than generating written words from image features, the Image2Speech system generates speech units (phonemes), which can then be synthesized into speech. Image2Speech circumvents the need for an orthography and it is therefore in principle applicable to any spoken language, provided that there is training material for it. An Image2Speech example can be seen in Figure 1.1 which shows the phoneme caption output that the Image2Speech system that has been developed in this research has generated for an example image from the Flickr8k dataset [37] (and the same caption manually converted into words).

Figure 1.1: An example image from Flickr8k [37] with generated caption:
EY D AO G IH Z R AH N IX NG TH R UW DH AX S N OW
(A dog is running through the snow)

1.1. Research Question

Because the Image2Speech task is new, no well-established method for evaluating the performance of a system for this task as yet exists. One of the main purposes of this research is to fill that gap. While human evaluation is an obvious option, it is also an expensive and time consuming method of evaluation. Not all researchers have the time and resources available for human evaluation. For cheap and quick evaluation there is the option of objective evaluation metrics. Many such metrics exist in the field of Natural Language

Processing (NLP), where they are often used to score the semantic similarity of two sentences, which can also be applied to evaluate automatic image captioning. Previous research into Image2Speech [16] has also made use of such metrics. These metrics are mostly designed for sequences (sentences) of words, however the Image2Speech system gives sequences of phonemes as its output. It is unknown how well currently existing objective metrics can evaluate the Image2Speech task. This research focuses on establishing that, which leads to the first and main research question (RQ1): **Which method of objective evaluation works best in determining the performance of an Image2Speech system?**

To answer this, the Image2Speech system needs to be evaluated with objective evaluation metrics and compared with human evaluation in order to establish how well the objective evaluation metrics correspond with the opinions of the end users. This leads to the sub research question (RQ2): *Which objective evaluation metric achieves the highest correlation with human evaluation?*.

Further, it is useful to know whether the output of an Image2Speech system is sufficient for producing comprehensible speech. The Image2Speech task aims to be applicable to unwritten languages, for which there are fewer types of resources that can be used for speech synthesis compared to written languages. This leads to the sub research question(RQ3): *Which resources does the Image2Speech system require in order to produce comprehensible speech?*

In order to answer these research questions the following has been done in this research:

- A new Image2Speech system is developed (see Section 3.1) in order to obtain outputs and evaluation results that can help answer RQ1 and RQ2. This system is heavily based on that of Hasegawa-Johnson et al. [16] and makes extensive use of deep learning. The evaluation (see Section 3.2) is mainly focused on how well the phoneme sequences generated by the Image2Speech system describe the image, in other words, how well the semantics of the image are represented in the phoneme sequences. This is because the main distinguishing factor between the Image2Speech system in comparison to most other image captioning systems is the direct conversion of an image representation into speech units. The new Image2Speech system uses English as its language, which is not an unwritten language. However it is treated as closely to an unwritten language as possible for the purposes of training the Image2Speech model by using phonemes as the desired output data.

- Image2Speech outputs are synthesized under multiple sets of circumstances, that reflect different scenarios of available resources in order to answer RQ3. Informal listening is done to determine in which scenarios comprehensible speech can be synthesized (see Section 3.1.4).

- The output of the Image2Speech system is evaluated by human raters with the use of crowdsourcing (see Section 3.2.1). The resulting human evaluation data is used to determine how well the Image2Speech system performs as well as to determine how well objective metrics can evaluate an Image2Speech system.

  - Since crowdsource workers can not be expected to be able to interpret sequences of phonemes very well, a simple method to convert phoneme sequences into words is developed to make the output readable and interpretable for the human raters (see Section 3.2.1).
  - To gain more insight into which aspects are most important to determine the goodness of the description of the caption, the raters are also asked about more specific aspects of the images, namely objects and actions, as these are likely to be the most significant aspects of a caption.

- The correlation between the results of the human evaluation and the results of several objective metrics (see Section 3.2.2) is determined using the Pearson correlation coefficient in order to establish which objective metric correlates the most with human evaluation and is thus best used to evaluate an Image2Speech system when human evaluation is not a feasible option.

Chapter 2 will talk about background knowledge related to Image2Speech. Chapter 3 explains the Image2Speech methodology and the evaluation methodology. Chapter 4 covers the results that have been obtained throughout this research. Finally Chapter 5 discusses the results, the implications and the limitations of this research.

2

Related Works

This chapter lays out the background information that is important for understanding the inner workings and evaluation of the Image2Speech system (Chapter 3), and also talks about related fields to the Image2Speech task. Deep Learning is covered in Section 2.1, Automatic Image captioning is covered in Section 2.2 and Crowdsourcing is covered in Section 3.2.1.

2.1. Deep Learning

Deep learning is a machine learning method based on Artificial Neural Networks (ANNs) that has become increasingly popular with the increased availability of data and processing power. Approaches that make use of deep learning have managed to achieve state-of-the-art results in fields such as image recognition[23], speech recognition[17][7], machine translation[11] and many other fields. Deep learning makes use of representation learning which are techniques that allow for extraction of information from raw data by transforming the data into a representation that is slightly more abstract. These abstract representations can be difficult for humans to interpret, however they are suitable for training machine learning models. Deep learning uses several layers of representation learning which leads to increasingly abstract representations. Typically this leads to very complex models even in comparison to other machine learning methods, however these models are often able to achieve better results than other models. Deep learning is very flexible and can be applied to a huge variety of tasks, however it often does require large amounts of data and its training process can take a very long time.

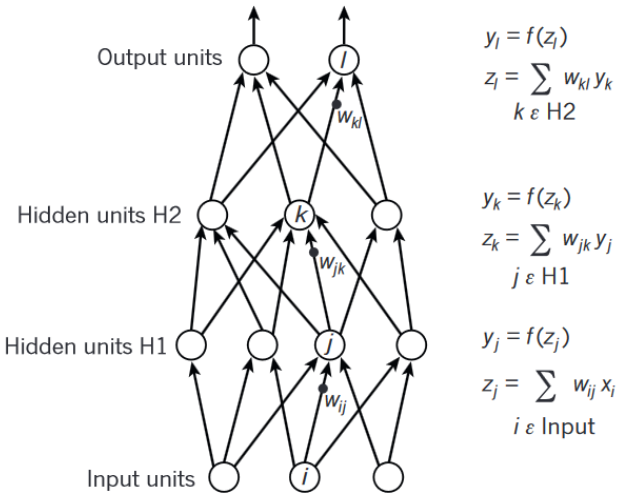

Figure 2.1: A feed-forward 4-layer neural network [25].

Many neural networks make use of a feed-forward structure which is visualized in Figure 2.1. This example neural network is represented as a graph with a number of nodes and connections. The nodes themselves represent non-linear functions that are applied to their input and the connections between nodes represent the flow of input and output data. At every node, its non-linear function is applied to the input and the result of that is given as an output. For example in Figure 2.1, the input node labeled *i* applies its non-linear function to the input and passes it to the node labeled *j* (as well as two other unlabeled nodes). This process of applying a function to the input and giving an output happens in every node. It starts at the input nodes which pass on their outputs to the next nodes and this process repeats until it reaches the output nodes, resulting in an output that is intended to be (close to) the desired output. As can be seen in Figure 2.1, the nodes form several groups which are called layers, starting with an input layer, ending with an output layer, and everything in-between is referred to as a hidden layer. The flow of data starts from the input layer, the input nodes pass their output forward through the connections to the nodes of the hidden layers until it reaches the output layer. A neural network can contain any number of hidden nodes and layers. When a neural network has multiple hidden layers it is often referred to as a Deep Neural Network (DNN). The non-linear functions contained in every node partially consist of a number of learned parameters, which are usually called the weights (w) and biases (b). Figure 2.2 shows how the weights and biases are applied to the input. The weights

are multiplied with their respective input and then the bias term is added. Afterward a non-linear activation function $f(\cdot)$ is applied. Commonly used non-linear include the ReLU unit ($f(z) = max(0,z)$, the sigmoid function ($f(z) = \frac{1}{1+e^{-z}}$ and the hyperbolic tangent function ($f(z) = tanh(z)$. Softmax [14]($f(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$) is an activation function that is commonly applied to the output of an architecture for multi-class classification, where it can be used to compute for every class a probability of a sample belonging to that class.

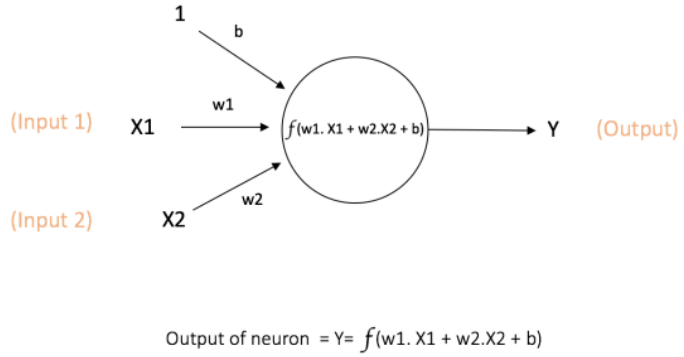

Figure 2.2: A node of a neural network with two inputs and one bias term.[20]

As with most machine learning methods, deep learning has a training process in which the parameters of the nodes are learned. There are different approaches to learning, which often depends on the resources that are available. Supervised learning is the most common approach which makes use of data that consists of input (features) and desired output (labels) data. A trained model is able to take features as an input, process them and give an output. Ideally the output of a trained model is the same as the desired output, or close to it, depending on the application. During training, the goal is to obtain the best score which is determined by something called the objective function. The objective function compares the output of the model with the desired output in order to determine a score. This score gives an indication of how well the model performs on the training data. The model that achieves the best score is not necessarily the best possible model, however it is the best model according to the automatic objective evaluation scheme of optimizing the objective function.

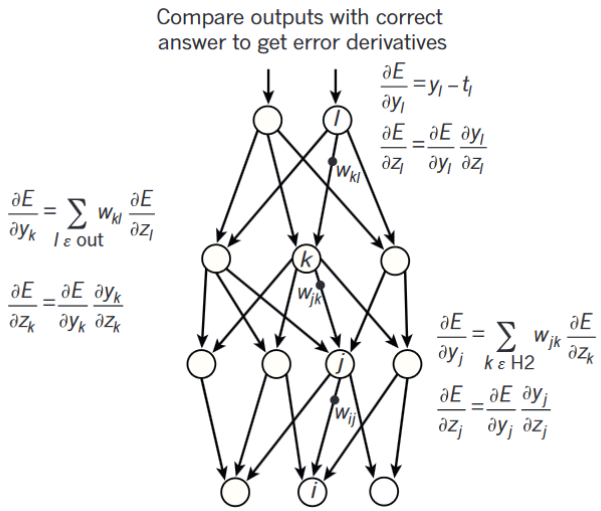

Figure 2.3: Backpropagation with the network shown in Figure 2.1 [25].

There are many methods for optimizing the objective function, one very popular one being Stochastic Gradient Descent (SGD). SGD involves calculating the gradients of the objective function. These gradients indicate whether a parameter should increase or decrease in value in order to obtain a better score. Figure 2.3 shows a method for calculating the gradients and tuning the weights, called backpropagation. At the

output layer the gradients of the objective function with respect to the output of the nodes are calculated by differentiating the objective function. These gradients are then used to calculate the gradients of the objective function with respect to the weights of the output nodes. Those gradients are passed on to the previous layer in the network in order to calculate the gradients with respect to the weights of the nodes in that layer. This process goes on until all necessary gradients are calculated so that the parameters can be tuned.

Unsupervised learning is different in that it only makes use of unlabeled data. This can be helpful since the process of labeling data can be expensive and time-consuming. However due to the absence of labels in the data, it can be difficult to train a model that targets the output that is desired. Unsupervised learning focuses more on learning the distribution and structure of the data. A common method of unsupervised learning called clustering aims to split the data into groups such that similar data points reside in the same group. Another example are autoencoders which learn a representation that can encode data which reduces the dimensionality, resulting in fewer features and less noise than the original data. Additionally an autoencoder also learns a reconstruction component that reconstructs a representation that is close to the original data.

Reinforcement learning work with a dynamic environment, which could for example be the real world or a simulation. Training is done with a trail and error approach, while receiving feedback signals from the environment. Reinforcement learning can be done without any prior data, however it is usually a very long process.

Besides the feed-forward networks there are many other variants of neural networks. Two prominent ones that are used in this research being convolutional neural networks and recurrent neural networks.

2.1.1. Convolutional Neural Networks

Since this research is focused on images, extensive use is made of CNNs. Convolutional Neural Networks (CNNs) are a very popular type of neural network for tasks that involve images. This is due to having its own feature extracting mechanism with convolutional layers and dimensionality reduction with pooling layers [25].

An example CNN architecture can be found in Figure 2.4 that is used to classify hand-written digits. CNNs are composed of convolutional layers and pooling layers in addition to a feed-forward network. As can be seen in Figure 2.1.1 the input enters several convolutional layers and pooling layers until it reaches the final feed-forward layers.

In the convolutional layers, the input is filtered by convolutional kernels (also known as filter banks) to create feature maps. Figure 2.5 shows a 5x5 input image being convolved with a 3x3 kernel, which results in what is called a feature map (shown right). The kernel is laid over the image and the values of the kernel (which are learned) are multiplied with the values of a 3x3 (the size of the kernel that is used) part of the image, which is then summed resulting in a value of 51 in the feature map. The kernel slides over the image with a predetermined step size until it has covered the whole image. Multiple kernels with different parameters are typically used in one convolutional layer, resulting in a set of feature maps per layer. These feature maps are typically a smaller size than the input to the convolutional layer, which means that the dimensionality has been reduced. Every filter strides over the entire image which helps make CNNs translation invariant. This means that they are not overtrained on the exact positions of certain features, e.g. only being able to recognize an object if it is positioned on the left side of an image.

Next, a non-linear activation function is applied to the feature maps to introduce non-linearity. Without non-linearity being introduced, the convolutions would be fully linear which heavily limits the complexity and performance of a CNN. After that they are passed to the pooling layer. The pooling layer further reduces dimensionality by aggregating the inputs that it gets. The aggregation reduces the size of the input and number of parameters which helps make them more manageable. Pooling layers are different from convolutional layers, because the aggregation (usually) does not involve convolution or learned parameters. Max pooling is the most popular type of pooling which simply takes the maximum value from its inputs, average pooling is also fairly common which averages the values of its input. Figure 2.6 shows an example of max pooling and average pooling. The input on the left is divided into groups, indicated by the colored boxes. On the right is shown what the output would be after max pooling or average pooling is applied to the groups of the input.

A typical CNN structure contains multiple convolutional layers stacked on each other with pooling layers in-between them. CNNs have obtained state-of-the-art results for many tasks, particularly tasks related to processing images such as object recognition[40], image classification[26] and facial recognition[19].

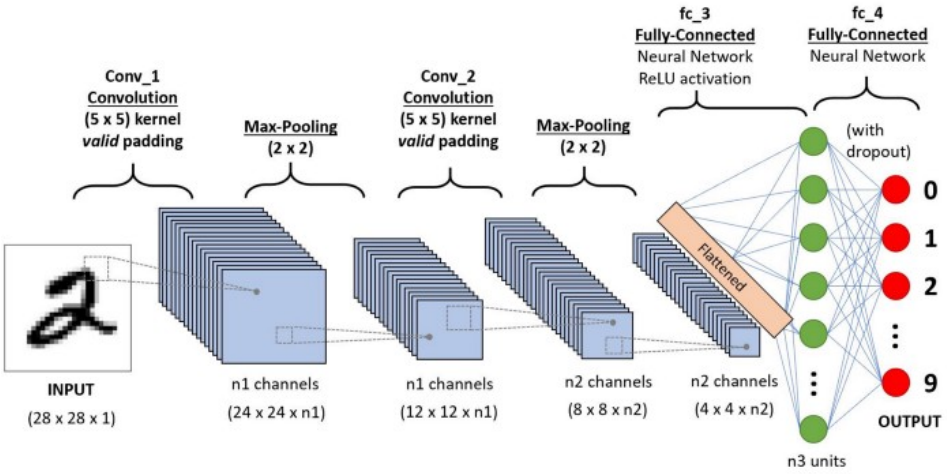

Figure 2.4: A CNN that classifies hand-written digits[38].

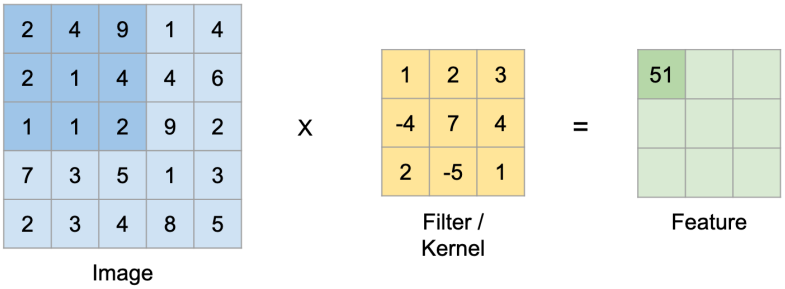

Figure 2.5: A convolutional filter being applied to an input image to fill in a feature map[35].

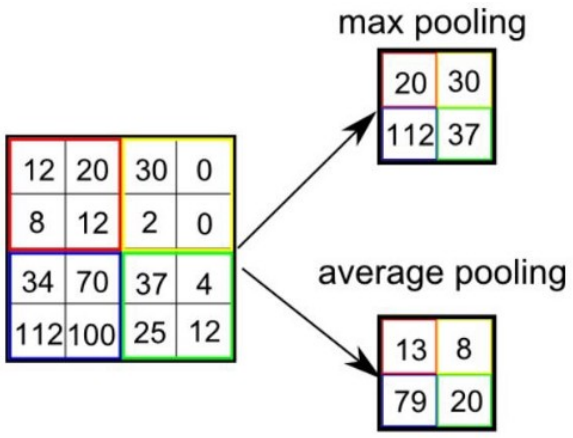

Figure 2.6: Max pooling and average pooling of a 4x4 input[38].

2.1.2. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of neural network that work particularly well for sequential data and/or sequential output, such as generating sequences which is part of the main goal of Image2Speech. The main distinguishing feature of an RNN compared to most other neural network types is its looping structure, which can be seen in Figure 2.7 which shows the basic structure of an RNN. This looping structure allows an RNN to take a sequence of multiple inputs as well as give a sequence of multiple outputs. The first input x_0 is passed to a node which is more complex than a typical neural network node. This node contains a structure of neural network layers and operations, which as a whole shall be referred to as A. A processes x_0 and gives an output h_0. Then the next input x_1 is passed to the next node which contains the same neural network A, however this node also receives an input from the previous node. The input of previous nodes acts as a sort of memory while processing the sequence $x_0, ..., x_t$ which helps retain information about previous inputs. By retaining this information, RNNs are very well suited for NLP tasks. For example generating a sentence word for word requires looking back at previous outputs in order for the sentence to make sense both grammatically and semantically. RNNs can be applied to a variety of problems such as those that involve predicting one output per input that is part of a sequence, or predicting a single output from a whole sequence. However there are also problems where the length of the input and output can both vary and not be the same, called sequence-to-sequence problems, for which RNNs are not sufficient. For these problems, RNNs can be used in a encoder-decoder architecture[6], which makes use of two RNNs. One RNN acts as the encoder which encodes an input sequence of varying length into an output sequence of a fixed length. The other RNN acts as the decoder, which decodes an input of fixed length into an output of varying length. The encoder-decoder architecture is also used for the Image2Speech system developed in this research which encodes image features and decodes with phonemes as an output.

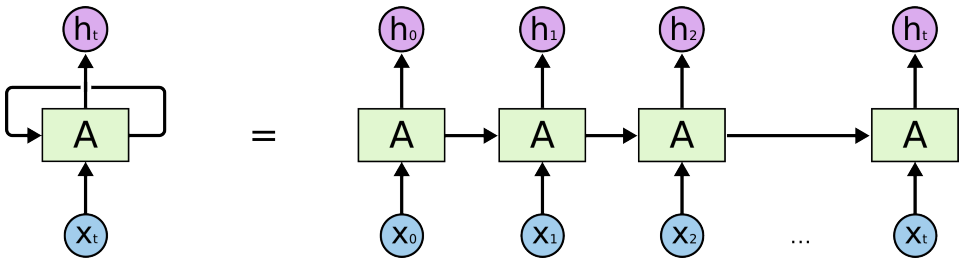

Figure 2.7: The structure of an RNN.[30]

Long Short-Term Memory

One of the downsides of a regular RNN is how it has trouble retaining important information after it has gone through many nodes. This is particularly a problem for Image2Speech which generally has to give a relatively long sequence of outputs since it takes multiple phoneme outputs to make up one word. A solution to that is the Long Short-Term Memory (LSTM) network. This is an RNN variant which uses a more complex node structure in order to retain important information and discord (forget) unimportant information. Figure 2.8 shows what a typical RNN node looks like, which contains one tanh layer which takes input from the input sequence and from the previous node. Figure 2.9 shows what an LSTM node looks like, which is more complex and requires extensive explanation.

The LSTM node has 4 main components:

- **The memory** which in Figure 2.9 is represented by the upper line. The memory carries information from previous nodes which is combined with the input to produce an output.

- The three gates which are represented as yellow blocks in Figure 2.9, and described in order from left to right:

  - **The forget gate** is where part of the information in the memory is discarded which is not needed anymore. This happens through a sigmoid layer, which maps input values (taken from the current input x_t and the previous output h_{t-1}) to output values between 0 and 1. The memory's values are multiplied element-wise with these values, which means that a low value causes information to be discarded and a high value causes it to be retained.

– **The input gate** is where new information from the current input is added to the memory's values. This gate consists of two layers, another sigmoid layer and a tanh layer which maps input values to output values between −1 and 1. The outputs of these layers are multiplied element-wise and then added element-wise to the memory's values. This way the sigmoid layer determines which values are changed and the tanh layer determines how those values are changed.
– **The output gate** is where an output is generated. This gate involves yet another sigmoid layer, the output of which is multiplied element-wise with the memory's values with a tanh function applied to it. The node then gives as an output h_t and passes the memory and h_t to the next node. [30]

LSTMs have been able to achieve state-of-the-art results for several NLP tasks, such as automatic image captioning[13]. They are also often used as the RNNs in the encoder-decoder architecture[46].

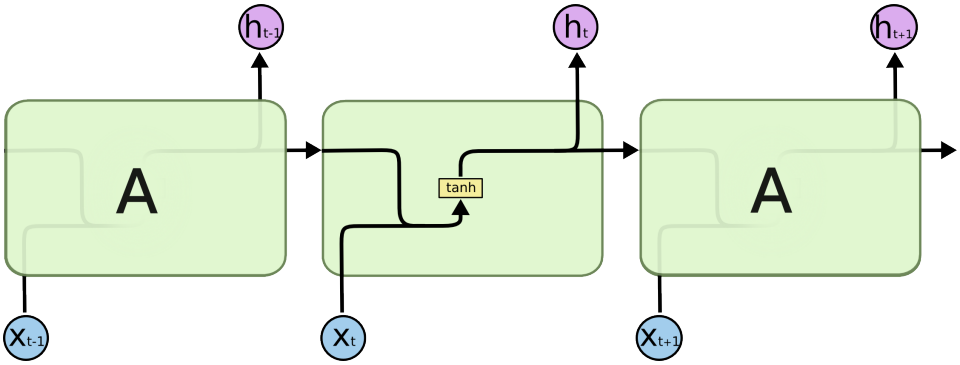

Figure 2.8: The structure of a typical regular RNN node.[30]

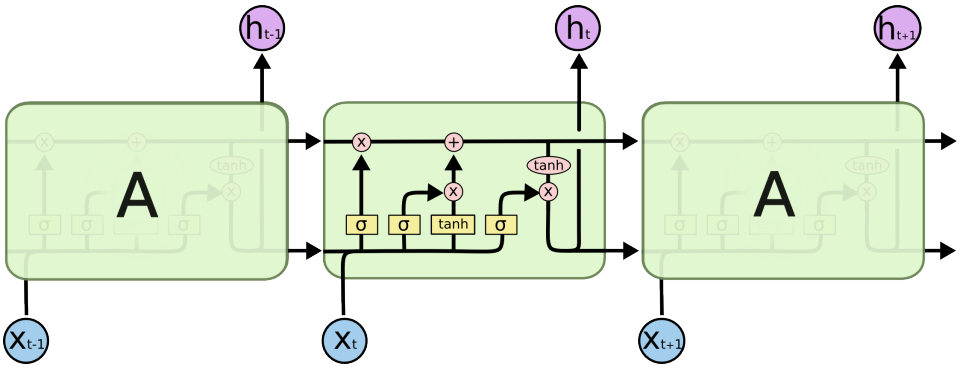

Figure 2.9: The more complex LSTM structure.[30]

There are also many variants of LSTM networks, such as the bi-directional LSTM (BLSTM) which does not only take past inputs into consideration, but future inputs as well. Using BLSTMs in a pyramidal structure allows for output sequences that are shorter than the input sequence. Figure 2.10 shows this structure where the number of nodes is halved with each layer, causing the number of output nodes to be only a quarter of the number of input nodes. This method is useful for encoding long sequences, such as a sequence of images features which is used in this research. With a regular LSTM it is difficult to extract relevant information large inputs due to the large amount of information. BLSTM also helps speeding up the time it takes to train a model by reducing the size of the input.

2.2. Automatic Image Captioning

Automatic Image Captioning is the task of automatically generating a description (caption) of an image[33] which Image2Speech [16] is based off. While Image2Speech generates *spoken* captions, almost all existing Automatic Image Captioning systems generate *textual* captions, essentially these systems are Image2Txt systems. Figure 2.11 shows this hierarchy, with Image2Txt and Image2Speech being sub-tasks of Automatic Image Captioning. The main difference between these tasks is that the format of Image2Txt is a textual caption, while for Image2Speech it is a phoneme caption which is synthesized into a spoken caption.

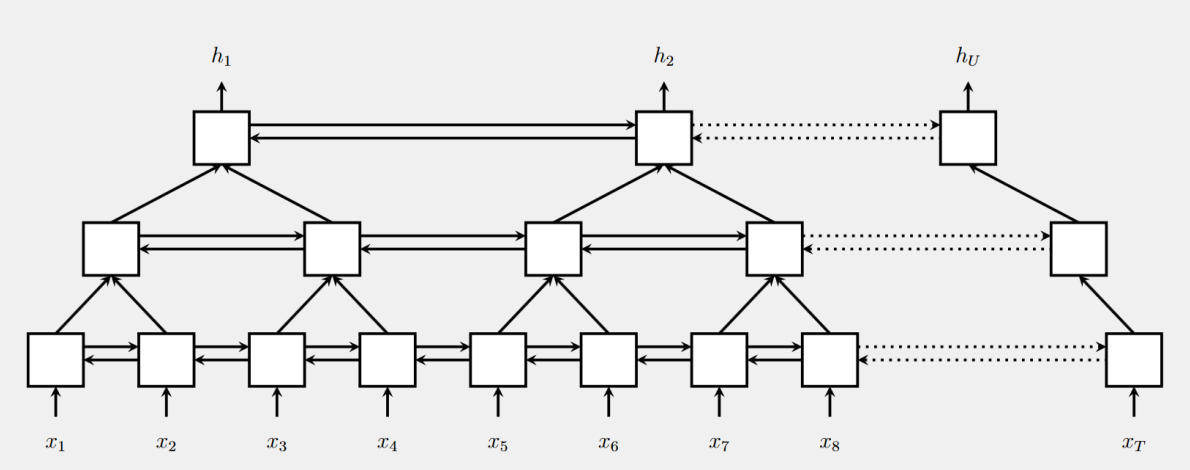

Figure 2.10: Pyramidal BLSTM structure[4].

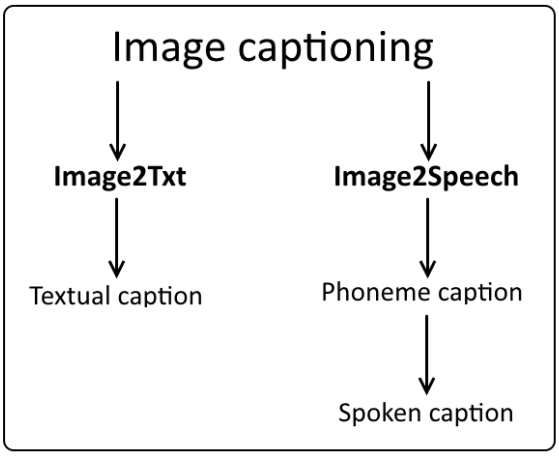

Figure 2.11: Automatic Image Captioning and its sub-tasks

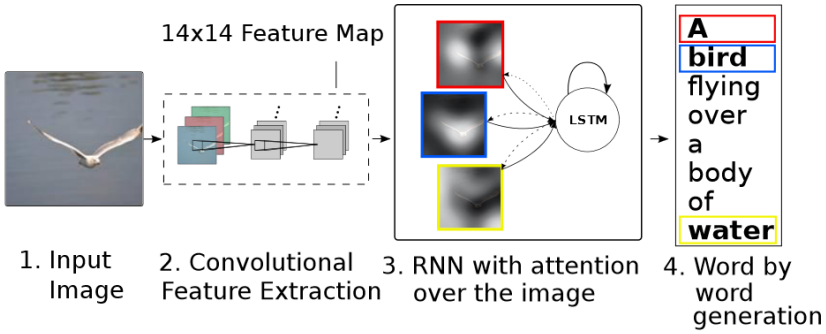

Figure 2.12: Image2Txt by [46].

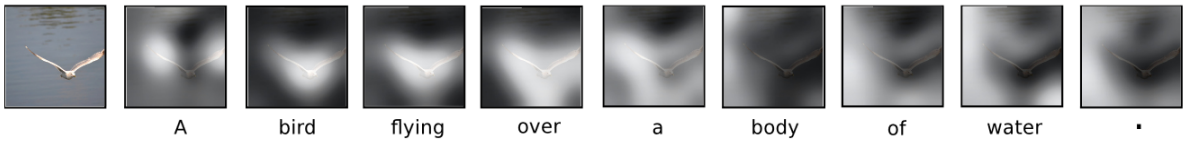

Figure 2.13: Visualized attention of an Image2Txt system[46]. White regions indicate high attention.

Image2Txt systems are generally very similar to an Image2Speech system, since the tasks are very similar as well. There are mainly two differences between an Image2Txt system and an Image2Speech system. First of all the output for an Image2Txt system is a written caption, meaning that there is no need for any speech synthesis. Second, the captions that the system trains on and outputs are made up of words instead of phonemes, which means that it relies on textual resources.

Figure 2.12 shows a common approach for an Image2Txt system. It starts with an input image. Then a CNN(Section 2.1.1 is used to extract features from an input image. These image features represent the image in a way that reduces the dimensionality of the input while still keeping the most important aspects of the image. The next step is for the image features to be encoded (see Section 2.1.2 for the encoder-decoder architecture) and finally decoded to generate a sequence of words.

Image2Txt architectures also often contain an *attender* component. An attender dynamically places emphasis on parts of the input in order to make the relevant parts of the image more salient. For the task of Image2Txt, attention can be visualized to show which areas of the image are being looked at by the system when outputting a word. Figure 2.13 shows an example[46] of this where for every outputted word the white parts highlight where the relevant parts of the image are. Xu et al.[46] shows how using attention can be very beneficial in Image2Txt and achieved state of the art results at the time with it.

Image2Txt is usually evaluated with the objective metric BLEU (bilingual evaluation understudy) [34] and often METEOR] (Metric for Evaluation of Translation with Explicit ORdering) [3] is included as well[46][12][9], which can give an indication on the semantic similarity between the generated caption and the ground truth. Two other objective metrics, ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) [27] and CIDEr (Consensus-based Image Description Evaluation) [43] have also recently been used as an evaluation method[13], however they are not used very commonly.

2.3. Crowdsourcing

Crowdsourcing is a method that uses a large group of individuals to performs Human Intelligence Tasks (HITs) in return for monetary compensation [22]. There are many possible applications for crowdsourcing such as evaluation, data annotation and surveys. It is also used for the human evaluation done in this research. Crowdsourcing is well suited for tasks that would take a large amount of time and effort for one individual to complete, but can be divided up and delegated to multiple individuals in order to get the task done quickly. As an example the flickr8k corpus[37] which is used in this research is a dataset that has been created with the use of Amazon's Mechanical Turk, which is an online platform for crowdsourcing purposes. Flickr8k contains 8000 images which have been annotated by crowdsource workers, resulting in 5 textual descriptions of every image in the dataset. For a small team of researches it would take a lot of time and effort to annotate 8000 images 5 times each, but with crowdsourcing it can be done with relative ease. Crowdsourcing is not without its downsides however. Obviously there is a monetary cost which can be a barrier for

researchers. There is also the risk of unreliable crowdsource workers who do not perform the given tasks seriously or competently. It is possible to account for those issues in various ways such as giving the same task to multiple crowdsource workers or having part of the task be a hidden test[24] to help determine whether they are serious and competent.[18]

Crowdsourcing also brings various ethical concerns, particularly of note is that it can be seen as the exploitation of cheap labor [41]. Crowdsource workers can easily be underpaid because there are incentives to take on underpaying tasks. For example many crowdsource platforms keep track of the performances of workers as a form of credentials. Those that successfully perform many tasks get good credentials and may gain access to perform tasks with higher requirements, which typically provide higher compensation. This incentivises workers to perform underpaying tasks in order to build up their worker credentials. There is also a lot of competition among workers, as good paying tasks are usually very quickly taken. This again incentivises workers to perform underpaying tasks, since it can be difficult to find a good paying tasks before it is already taken by someone else. Ideally those who make use of crowdsourcing should consider the time it takes to perform their tasks and base the compensation off of that time, such that the hourly compensation is at least equal to the hourly minimum wage (of the country that most workers can be expected to reside in).

3

Methodology

This chapter lays out the methodologies used and consists of two parts. The first part explains the methodology of creating the Image2Speech system. The second part explains the methodology of how output of the Image2Speech system is evaluated.

3.1. Image2Speech system

A new image captioning system that is based on the Image2Speech system [16] has been developed, in order to obtain generated phoneme captions and also because the first system made by [16] is not publicly available. An overview of the system can be found in Figure 3.1. The first step of the system is to extract image features from the input with the VGG16 model (see Section 3.1.2. These image features are then used as input for the image-to-phoneme model to generate a caption consisting of phonemes (see Section 3.1.3). Lastly the phoneme sequences are synthesised into speech with an audio synthesis model (see Section 3.1.4), this model was not implemented for this research but instead a pre-trained model was used. The architecture of the Image2Speech system is further explained in this section.

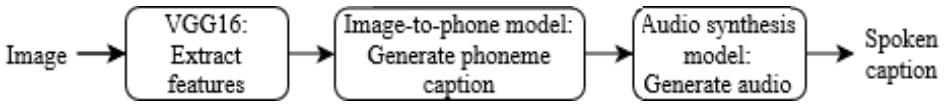

Figure 3.1: Image2Speech pipeline

3.1.1. Data

The datasets that were used were the Flickr8k [37] image and text caption corpus and its associated Flickr-Audio corpus [15]. The Flickr8k corpus contains 8000 images from Flickr, with five textual captions for each of these images, totalling 40,000 captions. The Flickr-Audio corpus contains recordings of each of the 40,000 captions being read aloud. Both datasets were created with the use of Amazon Mechanical Turk workers. The captions of Flickr8k also been converted into ARPABET phoneme sequences by Hasegawa-Johnson et al. [16] using the Janus Recognition Toolkit [21]. These images and phoneme sequences were used to train the image-to-phoneme model of the Image2Speech system. The training set of Flickr8k consists of 6,000 images with their phoneme captions, while the validation and test sets each contain 1,000 images and their captions (with no overlap). However due to the automatic phonetic transcription sometimes failing, the total number of captions used has dropped slightly and not all images have five phoneme captions available. This was mainly caused by out-of-dictionary words appearing in the textual captions. As this sometimes happened to all five captions of an image, meaning that those images could not be used. In total 5,956 images were used for training, 941 for validation and 959 for testing, with up to 5 captions per image totalling 28,205 captions for testing, 4,741 for validation, and 4,705 for testing.

3.1.2. Image Features

In order to train a model that uses images as input and generates a sequence of phonemes describing the image as output, feature extraction is required to condense the images into sets of features that are suitable for training a neural network. For this, the pre-trained VGG16 model is used, which is a convolutional neural network model (see Section 2.1.1) developed by Simonyan and Zisserman [40]. This model has been trained on ImageNet [8] which is a dataset that consists of over 14,000,000 images for roughly 22,000 nouns that come from WordNet [28]. It was the best model submitted to the Large Scale Visual Recognition Challenge 2014 (ILSVRC2014). On the task of Classification+localization with provided training data, it has managed to obtain a classification error of 0.07325[1]. Currently it is still one of the best performing models. The network architecture, shown in Figure 3.2, consists of 13 convolutional layers (colored black in Figure 3.2) and 2 fully connected layers (colored blue), and has been trained on ImageNet [8] which is a dataset that consists of over 14,000,000 images for roughly 22,000 nouns that come from WordNet [28]. In order to obtain image features from VGG16, the network has been cut off at the last convolutional layer, before the last pooling layer. The size of this layer is $14 \times 14 \times 512$, or 196 sequential feature vectors of dimension 512, with every feature vector representing a 40×40 window of the original 224×224 image. The reason for doing this is because after this point the network becomes a feed-forward network that is specialized in a different task than Image2Speech. The output of the last convolutional layer can be used as features for the Image2Speech task.

[1] http://www.image-net.org/challenges/LSVRC/2014/results

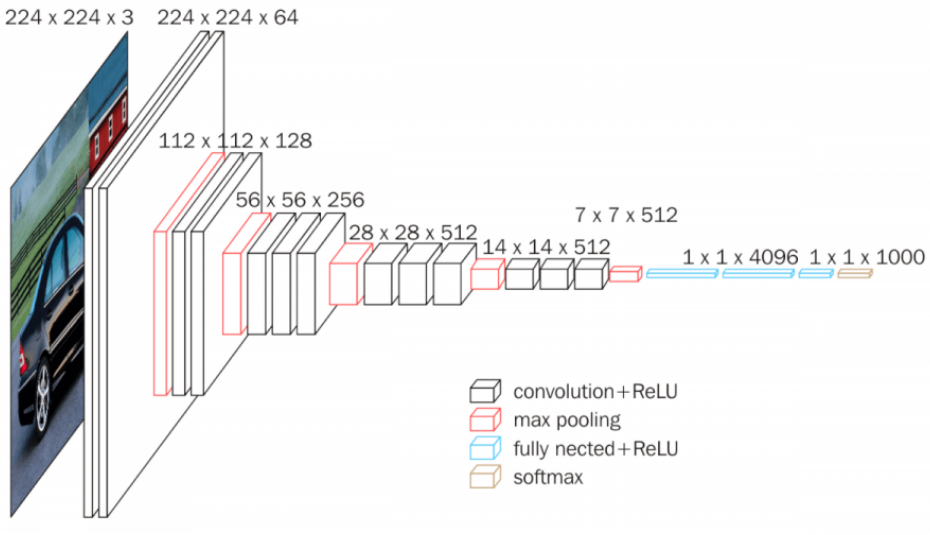

Figure 3.2: Architecture of the VGG16 network [42].

3.1.3. Image-to-phone Model

XNMT [29] (The eXtensible Neural Machine Translation Toolkit) has been used to train the image-to-phone model. XNMT is a neural network-based toolkit that is specialized in machine translation and general sequence-to-sequence modelling. The image-to-phone model is a sequence-to-sequence model, which is why the image features are represented as a sequence. This is done by keeping the order 196 feature vectors, representing 40×40 windows of the original images, as it is outputted by the final convolutional layer of the VGG16 model.

The image-to-phone model which is visualized in Figure 3.3 has 3 main components: an encoder, an attender, and a decoder. The encoder encodes the sequence of 196 feature vectors using a pyramidal LSTM (implemented with XNMT's PyramidalLSTMSeqTransducer) with 3 layers and a hidden dimension of 128. The attender applies attention to the output of the encoder using a multi-layer perceptron (XNMT's MlpAttender) with a state dimension of 512 and a hidden dimension of 128. The decoder is implemented with XNMT's default decoder which uses a one-directional LSTM with 3 layers (fully connected but some connections are omitted in Figure 3.3) and a hidden dimension of 512, and a multi-layer perceptron with a hidden dimension of 1024 between the LSTM and a final softmax layer. XNMT's default objective function was used which calculated the maximum likelihood loss. The main changes from the original system's architecture [16] are an increase of the encoder layers from 1 to 3 and an increase of the attender state dimension from 128 to 512. This increase in parameters has been observed to improve the results of the model.

The model is trained using the sequential image features as input and phonetic transcriptions of its captions as labels/output. While up to 5 captions per image are available, XNMT does not have an inbuilt functionality that can take multiple captions into consideration during training. Instead every image is paired up once with each of its captions, resulting in 5 training data points for an image that has 5 captions. The model was trained with an Adam optimizer, learning rate of 0.001, dropout of 0.1, batch size of 21 and running for 10 epochs. The BLEU4 metric as outputted by XNMT has been a guiding metric in the process of optimising the Image2Speech architecture.

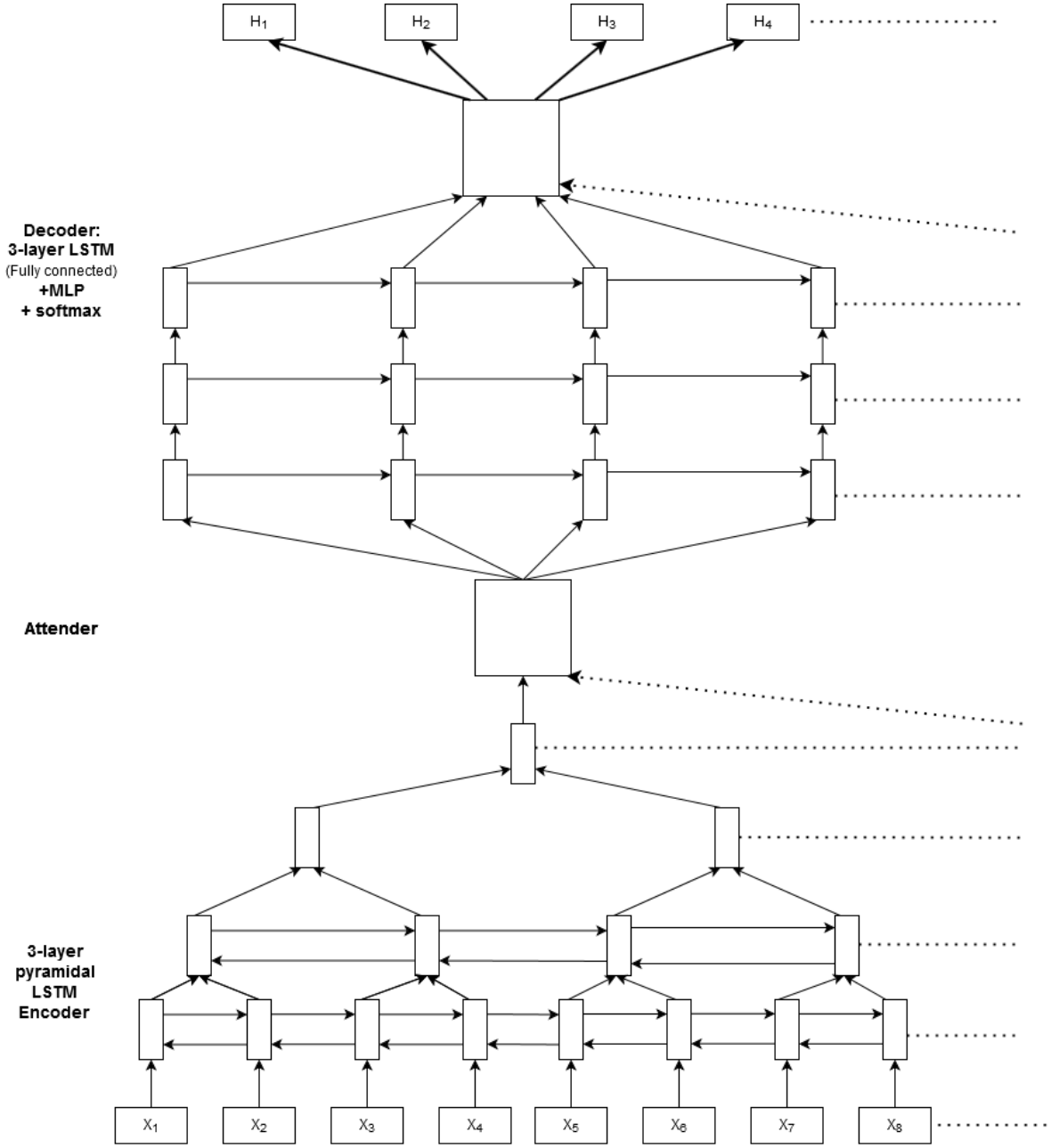

Figure 3.3: Architecture of the Image2Speech system. Inputs X are fed into the network resulting in outputs H.

3.1.4. Speech Synthesis

Speech synthesis has not been implemented in this research and instead samples have been synthesized by using a pre-trained speech synthesis model. The speech synthesis model that was used is a Tacotron-2 model [39] that has been pre-trained with phonemes as an input and speech as the desired output. Tacotron-2 is a text-to-speech system that uses an encoder-decoder architecture with attention to predict mel spectograms. A spectogram is a representation of the frequencies of an audio signal over time. A *mel* spectogram is a spectogram with frequencies converted to the mel scale. On this scale, an equal distance in pitch sounds equally distant to a listener. These mel spectograms are then converted into audio samples using an architecture based on WaveNet [31], which is a neural network system that is well-known for generating audio, including speech.

Speech samples have been synthesized from phoneme captions generated by the image-to-phone model. The synthesis have been done under multiple sets of circumstances. First if the presence of word boundaries, which indicate which phonemes represent a word. Second is the presence of linguistic stress, which places audible emphasis on certain phonemes. This results in four scenarios:

- No word boundaries, no linguistic stress.
- With word boundaries, no linguistic stress.
- No word boundaries, with linguistic stress.
- With word boundaries, with linguistic stress.

Word boundaries and linguistic stress are not generated by the Image2Speech system and instead have been added manually in order to determine how significant these factors are and whether that should be taken into consideration for future research into Image2Speech.

3.2. Evaluation

The original Image2Speech system was evaluated using two metrics: BLEU and PER (explained in Section 3.2.2). However it is unknown how well these metrics can evaluate an Image2Speech system, as they were not designed to be used for sequences of phonemes. This section describes the methods that are used to evaluate the Image2Speech task. Starting with subjective evaluation methods which uses human evaluation, and ending with objective evaluation methods which can automatically evaluate a system.

3.2.1. Subjective Evaluation

Subjective evaluation uses human evaluation to determine the performance of a system. In this research it is correlated with objective evaluation metrics in order to determine how well they can evaluate an Image2Speech system. The human ratings were collected with the use 409 of Amazon's Mechanical Turk (MTurk) workers, for monetary compensation ($0.60 per 30 image/caption pairs). The Human Intelligence Tasks (HITs) for this experiment have been set up using the output of the iteration that performed best on the BLEU4 metric.

Crowdsourcing

The test set of Image2Speech had 952 test image/caption pairs with succesfull phonetic transcriptions. These 952 image/caption pairs which were divided into 34 lists of 28 pairs without any overlapping pairs. Additionally, each list contained two control image/caption pairs with made-up captions: one image had a very bad caption and one image had a very good caption, which made for a total of 30 image/caption pairs per list. The control image/caption pairs were used to filter out raters who deviated too much from what was expected, e.g., due to a misunderstanding of the task. Every HIT contained one list of 30 image/caption pairs to be evaluated and every HIT was evaluated by five different evaluators. Three separate experiments have been run in order to determine how well the Image2Speech system performs according to humans. In experiment 1 the participants were shown 30 images with a caption and were asked to rate how well the caption described its corresponding image on a scale ranging from 1 (Very bad) to 7 (Very good). Experiments 2 and 3 were similar but instead asked the raters to rate how well the caption described the objects or actions, respectively, in the image on a scale from 1 (Very bad) to 4 (Very good). A smaller scale was used since there is less nuance involved when focusing on only one aspect. Prior to taking part in the HIT, the raters were provided with a number of example image/caption pairs from both ends of the rating scale to help them understand how

to interpret the scale. Raters were able to evaluate multiple lists and participate in multiple experiment but could not rate a list that they had already evaluated. Raters were compensated with $0.60 for every HIT, which on an hourly basis is roughly equivalent to the minimum wage of Amazon workers.

Phonemes to text
Crowdsource workers are very unlikely to be able to read caption sequences made up of phonemes. For this reason the captions are converted from phonemes sequences into sequences of words. This conversion is done with a simple weighted Finite State Transducer (wFST), implemented with OpenFST [1]. The wFST is a weighted graph with circular paths for every word where every node of a word's path represent the phonemes of that word. The wFST is created using a lexicon containing phonetic transcriptions of the words used in the flickr8k corpus. The weights of a graph are such that the total weight of a word exponentially decreases based on how many phonemes it is comprised of. As a result the total weight of a *word* is halved for every phoneme it contains. This is done in order to prevent small words from being giving priority over large words which contain smaller words. The wFST takes as an input the phoneme sequences generated by the Image2Speech model and gives as an output a sequence of words by finding the shortest path through the graph. This method does not make use of grammatical knowledge and is therefore prone to mistakes which mostly concern ambiguous cases such as similar sounding words (e.g. to/two). These mistakes have been corrected manually. A visualization of the wFST with many omissions is shown in Figure 3.4. It only contains the word "dog", as more words would add a lot of clutter. In the graph the '<s>' represents the start of a sequence, '</s>' represents the end of a sequence, '<eps>' represents an empty string and '<unk>' represents unknown characters. In this example you can see that there is a path from node 0 to node 4-6 and back to node 0 for the inputs "D", "AO" and "G" and the output "dog".

It is important to note that normally there are no textual resources used in the Image2Speech task, which this method obviously uses. The reason that textual resources are still used in this research is only to establish a relationship between objective metrics and human evaluation.

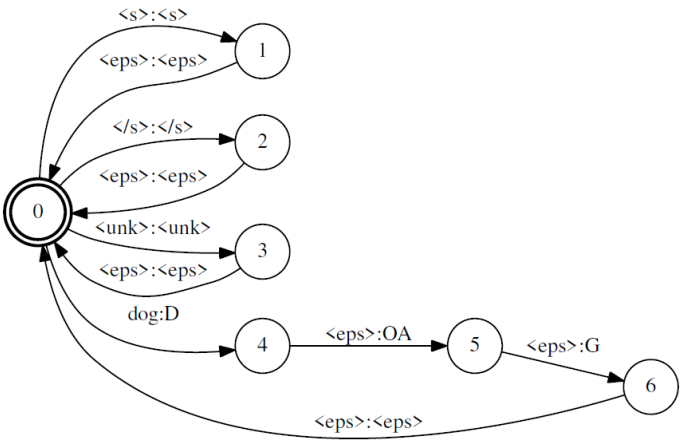

Figure 3.4: Visualization of a very simplified version of the wFST

3.2.2. Objective Evaluation

Objective evaluation metrics are automatic metrics which can give an indication on the performance of a system. The objective evaluation metrics that will be considered are all suitable for several NLP tasks. These metrics compare a hypothesis caption with one or multiple reference captions and give a score as an output that indicates the similarity or dissimilarity between the hypothesis and the reference(s).

Popular objective evaluation metrics for NLP tasks have been considered and are described below.

PER
Phoneme Error Rate (PER) (based on the Word Error Rate [45]), is a common metric used to determine the performance of ASR systems. It first aligns the hypothesis and reference sequences and then computes a score based on the number of substitutions, insertions and deletions of phonemes that are needed to create the reference out of the hypothesis.

The score is computed as:

$$\text{PER} = \frac{S + I + D}{N} * 100\% \tag{3.1}$$

where S is the number of substitutions, I is the number of insertions, D is the number of deletions and N is the number of phonemes in the reference sequence.

BLEU

BLEU [34] (bilingual evaluation understudy) is one of the most popular metrics for evaluating machine translations, it was developed as a fast metric for evaluating machine translations from one natural language to another.

BLEU makes use of a modified precision (shown below) of n-grams. N-grams are sequences of consecutive phonemes (in this context) of length N and are used very commonly in objective metrics. A number is often used to indicate the order of n-grams that are used, e.g. BLEU3 uses n-grams of up to order n=3. When there is no such indication, BLEU usually refers to BLEU4 which is the most commonly used BLEU score.

The modified n-gram precision p_n is computed as:

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{\text{n-gram} \in C} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{\text{n-gram}' \in C'} \text{Count}(\text{n-gram}')} \tag{3.2}$$

Where the $\text{Count}_{\text{clip}}$ is the minimum between the number of occurrences of the n-gram in the hypothesis sequence and the maximum number of occurrences of the n-gram between all reference sequences.

The BLEU-N score, with N being the length of the maximum length of the n-gram, is computed as:

$$BLEU - N = BP(exp(\sum_{n}^{N} \omega_n log(p_n))) \tag{3.3}$$

Where BP is a brevity penalty, and ω_n are uniform weights between the modified n-gram precisions p_n. The brevity penalty penalizes hypothesis sequences that are shorter than the reference sequence. With c as the length of the hypothesis sequence and r as the length of the reference sequence, the brevity penalty BP is computed as:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{otherwise} \end{cases} \tag{3.4}$$

ROUGE-L

ROUGE-L [27] (Recall-Oriented Understudy for Gisting Evaluation) is a metric designed for evaluating text summaries. While BLEU is precision based, ROUGE is as the name implies recall based. ROUGE-L makes use of the Longest Common Subsequence (LCS) between the hypothesis and reference sequences, i.e. the length of longest sequence of consecutive phonemes that appears in both sequences.

ROUGE-L for a reference sequence X of length m and hypothesis sequence Y of length n is computed by taking an F-measure of the LCS-based recall & precision:

$$R_{LCS} = \frac{LCS(X,Y)}{m} \tag{3.5}$$

$$P_{LCS} = \frac{LCS(X,Y)}{n} \tag{3.6}$$

$$ROUGE - L = F_{LCS} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{3.7}$$

With β being defined as:

$$\beta = P_{LCS} / R_{LCS} \tag{3.8}$$

when

$$\frac{\partial F_{LCS}}{\partial R_{LCS}} = \frac{\partial F_{LCS}}{\partial P_{LCS}} \tag{3.9}$$

CIDEr

CIDEr [43] (Consensus-based Image Description Evaluation) is a metric centered around comparing the similarity between a hypothesis sequence and a consensus of multiple reference sequences. It also makes use of n-grams up to n=4 and uses Term Frequency Inverse Document Frequency (TF-IDF) which assigns weights to every possible n-gram based on how frequently they appear in the corpus. TF-IDF assigns lower weights to n-grams that occur more frequently in the corpus, as those would be considered easy guesses.

CIDEr is computed with the following steps:

$$g_k(s_{ij} = \frac{h_k(s_{ij})}{\sum_{\omega_l \in \Omega} h_l(s_{ij})} * log(\frac{|I|}{\sum_{I_p \in I} min(1, \sum_q h_k(s_p q))}) \tag{3.10}$$

Where $h_k(s_{ij})$ is the number of times an n-gram appears in a reference sequence s_{ij} and $h_k(c_i)$ is the number of times an n-gram appears in the hypothesis sequence c_i.

Then a $CIDEr_n$ score, based on the cosine similarity, is computed for n-grams of length n:

$$CIDEr_n = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij}}{||g^n(c_i)||||g^n(s_{ij}||} \tag{3.11}$$

Where $g^n(c_i)$ is a vector formed by $g_k(c_i)$ corresponding to all n-grams of length n (similarly for $g^n(s_{ij})$).

Then for n-grams of several lengths n the scores are weighted and combined into one metric:

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n(c_i, S_i) \tag{3.12}$$

With N typically being set to 4 and weights being uniformly distributed, i.e. $w_n = 1/N$.

METEOR

METEOR [3] (Metric for Evaluation of Translation with Explicit ORdering)) is also popular metric for machine translation tasks.

METEOR starts with creating an alignment between the hypothesis and reference sequences. Then the precision and recall between the hypothesis and reference sequences is computed in the following manner:

$$RECALL = \frac{m}{w_t} PRECISION = \frac{m}{w_r} \tag{3.13}$$

Where m is the number of unigrams that are found in both the hypothesis and reference sequences. w_t and w_r are the number of unigrams in the hypothesis and reference sequences respectively.

The METEOR score is then computed as:

$$METEOR = F_{mean} * (1 - p) \tag{3.14}$$

Where F_{mean} is the harmonic mean between the precision and recall with recall weighing 9 times as much as precision:

$$F_{mean} = \frac{10 * PRECISION * RECALL}{RECALL + 9 * PRECISION} \tag{3.15}$$

And p is a penalty that depends on the number of *chunks* which are groups of unigrams that are adjacent in both the reference and hypothesis. This penalty exists in order to penalize sequences that do not align very well, i.e. do not have a similar order. Unigrams in the hypothesis and reference sequences are grouped into

the fewest number of chunks, such that the unigrams in every chunk are present and ordered in the same way for both the hypothesis and reference sequences. p is then computed as:

$$p = 0.5(\frac{c}{u_m})^3 \tag{3.16}$$

Where c is the number of chunks and u_m is the number of unigrams that have been mapped.

Type/Token Ratio
Type/Token Ratio (TTR) is a simple metric which is the number of unique words divided by the number of total words. This metric does not give an indication of the overall performance of the Image2Speech system but instead gives an indication on a system's lexical diversity.

4

Results

This chapter lists the results of the experiments performed. Experiment 1 which focuses on general caption quality and also a comparison with previous work is covered in Section 4.1. Experiments 2 & 3 which focus on specific aspects of images are put together in Section 4.2 as they are very similar and it allows for easy comparison. Section 4.3 covers an informal listening experiment on captions synthesized into speech.

4.1. Experiment 1: General Caption Quality

There is currently only one system to compare to which is the system developed by Hasegawa-Johnson et al. [16]. The BLEU4 and PER metrics are used for comparison since those are the only metrics that the Hasegawa-Johnson et al. published. Table 4.1 lists the obtained scores of both systems which shows the BLEU4 and PER scores of both systems. It is very important to note that the scores presented by Hasegawa-Johnson et al. were not computed the conventional way, which would be to compare hypothesis sequences to all available reference sequences at once. The scores in table 4.1 are the output from the XNMT toolbox. This method does not take into account that there are multiple captions for every image and instead treads every image/caption pair as separately. This is also why the scores of the new Image2Speech system are worse in comparison to Table 4.2. For a fair comparison, the scores in this table have been computed in the same way. As can be seen in Table 4.2 the new Image2Speech system has managed to obtain an improvement of 1.9 points on the BLEU4 score, however it obtained a worse PER score by 4.5 points.

Table 4.1: Comparison of old and new Image2Speech systems.

Metric	Hasegawa-Johnson et al. [16]	New Model
BLEU4	13.7	15.6
PER	84.9	86.4

The distribution of the results of the human evaluation regarding the overall quality of the captions can be found in Figure 4.1 which shows the number of ratings for every type of rating. The average overall score is 3.4 ($\pm$1.3 standard deviation), which would be between “Somewhat bad” and “Neutral”. Although the average results are on the low end of the scale, there is still a significant number of captions which have had ratings on the high end of the scale which indicates that the Image2Speech system does not produce only gibberish or random guesses. To give an indication of how a score relates to a captioned image, Figures 4.2 and 4.3 give an example of a very good caption (rated 6.4) and a bad caption (rated 2.0).

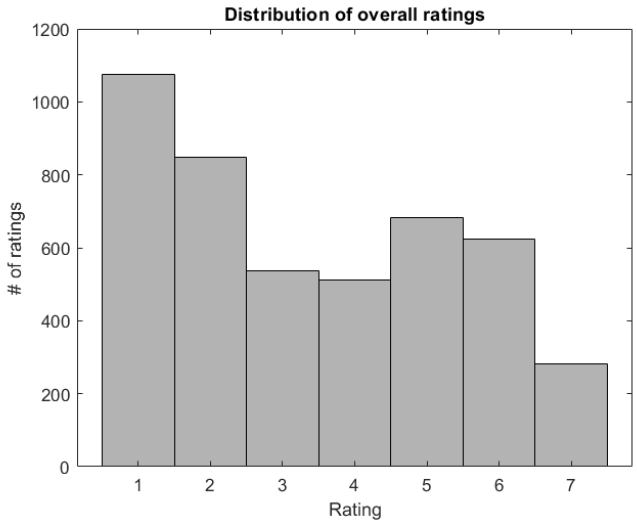

Figure 4.1: Distribution of overall ratings obtained from Amazon’s Mechanical Turk.

Figure 4.2: (rated 6.4) captioned: “EY G R UW P AX F S K IY R Z AXR S K IY IX NG D AW N EY S N OW IY HH IH L” (“A group of skiers are skiing down a snowy hill.”).

Figure 4.3: (rated 2.0) captioned: “EY M AE N IH N EY Y EH L OW SH ER T IH Z S T AE N D IX NG AA N AX S T R IY T” (“A man in a yellow shirt is standing on a street.”)

The scores of every metric and their correlation with human evaluation scores of overall quality can be found in table 4.2. Mturk scores are the Human evaluation scores of overall quality which can range from 1 (very bad) to 7 (very good). Objective metrics scores in the table can range from 0 to 100 with higher scores being better, with the exception of PER which is unbounded above and for which lower scores indicate a better performance. The individual scores of the objective metric have been correlated (using Pearson correlation) with human evaluation and these correlations are shown next to the scores. For every computed correlation it holds that $p < 0.001$. BLEU4 had the best correlation with the overall ratings, which corresponds to a weak to moderate correlation with the human ratings. BLEU5, BLEU3, ROUGE-L, and BLEU6 showed a weak to moderate correlation. BLEU7, BLEU2, BLEU8, PER, CIDEr, and METEOR only have a weak correlation. BLEU1 barely shows any correlation.

After converting the phonemes into words, the total number of words that was generated (tokens) was 11,060 and the number of unique words (types) was 255, making for a type/token ratio of 0.023. The ground truth, i.e., the textual captions of the corpus has a type/token ratio of 0.020. The output of the Image2speech system thus shows better lexical diversity than its training corpus.

4.2. Experiment 2 & 3: Action & Object Recognition

These experiments have been performed to gain more insight on which aspects of a caption are the most important. The results for the evaluations of actions and objects can be found in Figure 4.4 which shows the number of ratings which were generally on the low end. The average score for how well the actions are described by the captions is 2.1 ($\pm$0.8 standard deviation) and the average score for objects is 2.2 ($\pm$0.7 standard deviation), which in both cases roughly corresponds to “bad” on their scale. The difference in number of ob-

Table 4.2: Metric scores and correlations (r) with human evaluation (Mturk).

Metric	Score	**r**
Mturk	3.40	
BLEU1	82.6	0.155
BLEU2	61.3	0.355
BLEU3	46.4	0.425
BLEU4	36.1	**0.435**
BLEU5	24.6	0.429
BLEU6	18.2	0.410
BLEU7	13.7	0.378
BLEU8	9.3	0.340
METEOR	29.4	0.258
ROUGE-L	49.3	0.425
CIDEr	42.4	0.272
PER	71.4	-0.361

ject and action ratings is due to 81 HITs being rejected for the former. Feedback from the workers suggested that the control questions may not have been perfect indicators of whether a worker was competent and serious and for this reason HITs for action ratings were not rejected. Actions obtained a moderate to strong Pearson correlation of 0.57 ($p < 0.001$) with the ratings of overall quality and objects obtained a moderate to strong correlation of 0.63 ($p < 0.001$).

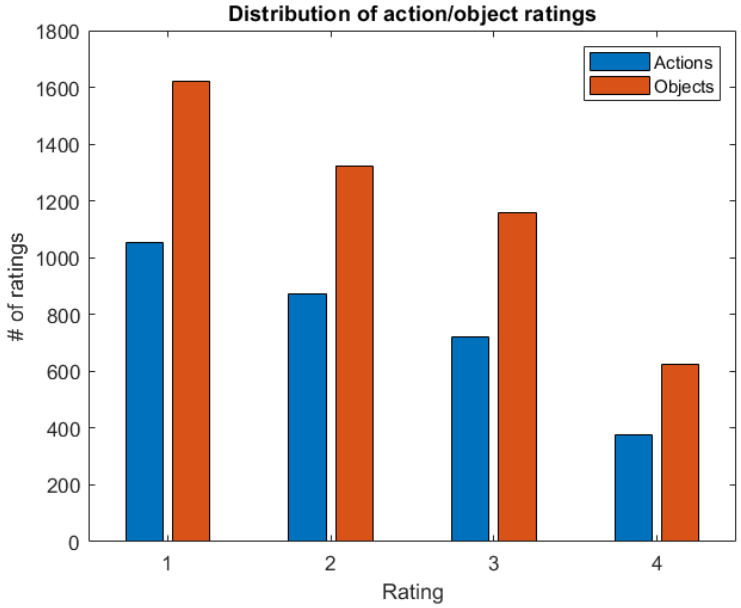

Figure 4.4: Distribution of action & object ratings obtained from Amazon's Mechanical Turk.

The actions and object ratings have also been correlated with the objective metrics to determine whether different metrics might capture different aspects better or whether these results will be similar the correlations in Table 4.2. The correlations of action and object ratings with objective metrics can be found in table 4.3 with the correlations for actions on the left and the correlations of objects on the right. Again for every computed correlation it holds that $p < 0.001$. The correlations for the action ratings are stronger for most metrics in comparison to the ratings of overall quality and even more so for the object ratings. BLEU4 remains the strongest correlating metric for every type of rating with a weak to moderate correlations for both types of ratings.

Table 4.3: Correlations of action & object scores with objective metrics and overall human evaluation scores.

Metric	$r_{actions}$	$r_{objects}$
Mturk	0.569	0.627
BLEU1	0.214	0.195
BLEU2	0.388	0.411
BLEU3	0.446	0.486
BLEU4	**0.449**	**0.494**
BLEU5	0.435	0.484
BLEU6	0.406	0.451
BLEU7	0.373	0.423
BLEU8	0.319	0.376
METEOR	0.265	0.322
ROUGE-L	0.416	0.486
CIDEr	0.305	0.315
PER	-0.363	0.381

4.3. Experiment 4: Spoken Caption Quality

Some captions have been synthesized into speech[1] using a pre-trained tacotron-2 model [39]. They have been synthesized in four different ways:

- No word boundaries, no linguistic stress.
- With word boundaries, no linguistic stress.
- No word boundaries, with linguistic stress.
- With word boundaries, with linguistic stress.

Ten phoneme captions with the highest ratings of overall caption quality were synthesized under these four scenarios. Informal listening has been done to evaluate these audio samples mainly focusing on whether they are comprehensible, meaning whether it was possible to tell which words were spoken. Samples without linguistic stress were not comprehensible at all, while those with linguistic stress were more comprehensible. Samples with linguistic stress and no word boundaries were comprehensible, however the word "a" would sometimes be omitted. Samples with word boundaries were comprehensible, had fewer (but still some) omissions of the word "a" and also sounded more natural, having better pauses between the words and placing emphasis on the right syllables more often.

[1] available at: https://github.com/Zoltandhaese/Image2speech

5

Discussion

This chapter discusses the interpretation of the results and the implications and limitations of the findings of this research.

5.1. Results Discussion

Phoneme captions have been generated by a new Image2Speech system using the Flickr8k dataset. These captions are evaluated with human evaluation and with objective evaluation metrics. The human evaluation and objective evaluation results are correlated in order to determine how well objective evaluation metrics can evaluate the Image2Speech task. The objective metric BLEU4 has consistently obtained the highest correlation with human evaluation, which means that it is currently the best existing metric to evaluate the Image2Speech task. It seems that metrics that use medium-length n-grams or Longest Common Subsequence are the most effective methods of evaluating Image2Speech in comparison to the other tested methods, however their overall correlation is still fairly low. METEOR and PER make use of sequence alignments instead and are not able to obtain a similar correlation as BLEU4 and ROUGE-L. CIDEr does use n-grams, however it also makes use of Term Frequency-Inverse Document Frequency, which likely works counterproductively. It assigns a lower weight to n-grams that are frequently occurring in the corpus, however most sentences have the same beginnings such as "A dog", "A man" or "A group of people" which are very important parts of a caption as they are often the subject. As stated earlier in Section 3.1.3 the BLEU4 metric has also been the leading metric that was used in the process of optimizing the model architecture to determine which architectures performed better. This might have introduced a bias in the correlations toward BLEU4. Unfortunately the correlation of 0.435 that BLEU4 obtained with human evaluation is not very high and BLEU4 is far from an ideal metric for the Image2Speech task. Ideally a metric for this task would be specifically designed with phoneme sequences in mind. Seeing as medium length BLEU and ROUGE-L obtain the highest correlations with human evaluation, it could be worth investigating whether a new metric using BLEU or ROUGE-L as a basis works well as a metric to evaluate Image2Speech. The new Image2Speech system has managed to obtain an average rating by human evaluators of 3.40 on a scale of 1 to 7, which is between "Somewhat Bad" and "Neutral". Clearly there is a lot of room for improvement for the system, but the ratings do show that the system is able to generate a significant amount of captions that describe their respective image well and does not output random gibberish. Ratings about actions and objects show a moderate to strong correlation with general caption quality. This shows that both actions and objects are indeed parts that a good caption should capture, with neither aspect being the sole indicator for overall caption quality. Objective metric scores generally appear to correlate more with solely action or object ratings than with overall caption quality ratings. This could indicate that the Image2Speech system currently is not performing well with other aspects of the image such as colors, backgrounds or aspects of the caption such as grammatical wellformedness.

Informal listening of synthesized speech samples which have been synthesized under four different scenarios, reveal that linguistic stress is very important in synthesizing comprehensible speech. While the addition of word boundaries does make the speech more natural, it does not seem to be crucial in making the speech comprehensible. It is fortunate that word boundaries are not crucial. Adding word boundaries would typically require textual resources and that could pose problems when Image2Speech is applied to unwritten languages. Linguistic stress can be learned by an Image2Speech system if it is contained in the training data, however the current phonetic transcriptions of Flickr8k unfortunately do not contain linguistic stress.

5.2. Dataset

While the Flickr8k dataset is in principle very well suited for a task such as Image2Speech, it also has its limitations. Deep neural networks work best with a large amount of training data and Flickr8k might not be large enough for this task. There are similar datasets which are much larger, allowing for more training samples, such as Flickr30k [36], Places [47] and MSCOCO [5]. Using a larger dataset would make the training process longer and might be less practical for some researchers, but it would likely lead to better captions due to having access to more training data. Flickr8k also has an issue of bias. While the dataset covers a lot of different kinds of scenes, there is a large imbalance in its diversity which is exemplified in Table 5.1. "Dog" is the most common noun to appears throughout all the captions and "Man" appears roughly twice as often as "Woman". This probably means that the Image2Speech model trained on Flickr8k performs a lot better for images that contain a dog than it does for images that contain a cat for instance. Ideally a dataset has a high and balanced diversity so that a model trained on it can will be less biased towards certain words.

Making improvements on the Image2Speech model was not the highest priority of this research and there is a lot of room for improvement. The new Image2Speech model is very similar to that of Hasegawa-Johnson

Word	# of occurrences
Dog	10263
Man	8296
Boy	4247
Girl	4169
Woman	4055

Table 5.1: The five most occurring nouns in the flickr8k captions.

et al. [16], with only an increase in the number of layers and nodes. It obtained a higher BLEU4 score, the best correlating metric, in comparison to Hasegawa-Johnson et al. which indicates that these increases resulted in a better model. The model could very well score higher with even more increases in the number of nodes, since increasing the number of nodes in comparison to Hasegawa-Johnson et al. resulted in a higher BLEU4 score. One potential major improvement to the training process that could be made is to take into account the fact that there are multiple captions per image. Currently the input is interpreted as image-caption pairs such that the system sees five different images with one caption each, instead of one image with five captions. This is not an optimal approach, because even if an output caption corresponds perfectly with one of the reference captions, it could still result in a high loss if it deviates too much from the other captions of its corresponding image. Changing the system so that the input is interpreted as five captions per image would almost certainly result in improvements.

6

Conclusion

This thesis presents an investigation on how to evaluate the relatively new task of generating captions made up of phonemes, from images, without the use of textual resources so that it can in principle be applied to unwritten languages. A new Image2Speech model has been trained on image/caption pairs from the Flickr8k dataset. The language of this dataset is English, which is a written language, however it has been treated as an unwritten language as much as possible for the purposes of training the Image2Speech model. Ideally, if the data is available, future research could apply the Image2Speech task to unwritten languages in order to test the assumption that the Image2Speech task can indeed be applied to unwritten languages. This model has achieved a better BLEU4 score than the model by Hasegawa-Johnson et al. [16]. Adding more complexity to the model by increasing the hidden dimensions and the number of layers seemed to be beneficial for the task. Nevertheless, the human ratings showed there is still a lot of room for improvement since the average rating was only a 3.4 out of 7. The current Image2Speech model has its shortcomings, such as the way it interprets its input during training, which can be improved on and there are many different existing methods, such as different attention models and decoders, which have not been tried that could potentially lead to further improvements in performance. Human evaluation of Image2Speech was obtained through Amazon's Mechanical Turk. Several Natural Language Processing metrics were then correlated with the human ratings in order to establish which metric is the best objective metric for Image2Speech. The BLEU4 metric obtained the highest correlation with the human ratings, closely followed by BLEU5, BLEU3 and ROUGE-L. This pattern was also found when more specific aspects (i.e., actions and objects) were rated instead of the overall quality of the caption. Correlations between metrics and ratings of specific aspects were generally stronger than between metrics and ratings of overall quality. It is notable that the metrics with the highest correlation make use of medium to high length n-grams (i.e., 3-grams, 4-grams, 5-grams and Longest Common Subsequence). Although BLEU4 obtained the highest correlation with human evaluation, it is not a strong correlation and is therefore not a perfect representation of human evaluation. At this moment however, BLEU4 is the best available indicator. Currently there is no metric that is specifically designed to determine the semantic similarity between an image and a sequence of phonemes. Informal listening reveals that the addition of linguistic stress is crucial for making comprehensible spoken captions. Speech samples synthesized without the use of linguistic stress were simply not comprehensible. Word boundaries were found to be of less importance. Speech samples without word boundaries were still comprehensible, however speech samples with word boundaries sounded more natural. Future research should take this into account and make use of (or create) a corpus with phonetic transcriptions that have linguistic stress.

Bibliography

[1] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. Openfst: A general and efficient weighted finite-state transducer library. In *Proceedings of the 12th International Conference on Implementation and Application of Automata*, CIAA'07, pages 11–23, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76335-X, 978-3-540-76335-2. URL http://dl.acm.org/citation.cfm?id=1775283.1775287.

[2] Shuang Bai and Shan An. A survey on automatic image caption generation. *Neurocomputing*, 311:291–304, 2018.

[3] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W05-0909.

[4] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *CoRR*, abs/1508.01211, 2015. URL http://arxiv.org/abs/1508.01211.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015. URL http://arxiv.org/abs/1504.00325.

[6] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

[7] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

[8] Jia Deng, Kai Li, Minh Do, Hao Su, and Li Fei-Fei. Construction and analysis of a large scale image ontology. *Vision Sciences Society*, 186(2), 2009.

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[10] David M Eberhard, Gary F Simons, and Charles D Fennig. Ethnologue: languages of the world. dallas, texas: Sil international. *Online version: http://www. ethnologue. com*, 2019.

[11] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *CoRR*, abs/1808.09381, 2018. URL http://arxiv.org/abs/1808.09381.

[12] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[13] Kun Fu, Junqi Jin, Runpeng Cui, Fei Sha, and Changshui Zhang. Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2321–2334, 2016.

[14] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[15] David Harwath and James Glass. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 237–244. IEEE, 2015.

[16] Mark Hasegawa-Johnson, Alan Black, Lucas Ondel, Odette Scharenborg, and Francesco Ciannella. Image2speech:automatically generatin gaudio descriptions of images. *Journal of the International Science and General Applications*, 1(1), 2018.

[17] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[18] Tobias Hossfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing. *IEEE Transactions on Multimedia*, 16(2):541–558, 2013.

[19] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster r-cnn. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 650–657. IEEE, 2017.

[20] Ujjwal Karn. A quick introduction to neural networks, 2016. URL `https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/`.

[21] Kevin Kilgour, Michael Heck, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. The 2014 kit iwslt speech-to-text systems for english, german and italian. In *International Workshop on Spoken Language Translation (IWSLT)*, 2014.

[22] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456, 2008.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, volume 2126, pages 22–32, 2010.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[26] Hyungtae Lee and Heesung Kwon. Going deeper with contextual cnn for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10):4843–4855, 2017.

[27] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-1013`.

[28] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[29] Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, Boston, March 2018.

[30] Christopher Olah. Understanding lstm networks, 2015. URL `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[31] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[32] Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos. Automatic image captioning. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 3, pages 1987–1990. IEEE, 2004.

[33] Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos. Automatic image captioning. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 3, pages 1987–1990. IEEE, 2004.

[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

[35] Krut Patel. Convolution operation— comprehensive guide, 2019. URL https://mc.ai/convolution-operation-comprehensive-guide/.

[36] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *CoRR*, abs/1505.04870, 2015. URL http://arxiv.org/abs/1505.04870.

[37] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.

[38] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way, 2018. URL https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[39] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[41] Susan Standing and Craig Standing. The ethical use of crowdsourcing. *Business Ethics: A European Review*, 27(1):72–80, 2018.

[42] Muneeb ul Hassan. Vgg16 – convolutional network for classification and detection, 2018. URL https://neurohive.io/en/popular-networks/vgg16/.

[43] Ramakrishna Vedantam, C. Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. pages 4566–4575, 06 2015. doi: 10.1109/CVPR.2015.7299087.

[44] Xinsheng Wang, Tingting Qiao, Jihua Zhu, Alan Hanjalic, and Odette Scharenborg. S2igan: Speech-to-image generation via adversarial learning. *ArXiv*, abs/2005.06968, 2020.

[45] Ye-Yi Wang, Alex Acero, and Ciprian Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*, pages 577–582. IEEE, 2003.

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. URL http://arxiv.org/abs/1502.03044.

[47] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.