FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Jindřich Libovický

Multimodality in Machine Translation

Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Pavel Pecina, Ph.D.

Study Program: Computer Science
Specialization: Computational Linguistics

Prague 2019

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

Prague, March 21, 2019 Jindřich Libovický

Title: Multimodality in Machine Translation

Author: Jindřich Libovický

Department: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Pavel Pecina, Ph.D.,
Institute of Formal and Applied Linguistics

Abstract:

Traditionally, most natural language processing tasks are solved within the language, relying on distributional properties of words. Representation learning abilities of deep learning recently allowed using additional information source by grounding the representations in the visual modality. One of the tasks that attempt to exploit the visual information is multimodal machine translation: translation of image captions when having access to the original image.

The thesis summarizes joint processing of language and real-world images using deep learning. It gives an overview of the state of the art in multimodal machine translation and describes our original contribution to solving this task. We introduce methods of combining multiple inputs of possibly different modalities in recurrent and self-attentive sequence-to-sequence models and show results on multimodal machine translation and other tasks related to machine translation. Finally, we analyze how the multimodality influences the semantic properties of the sentence representation learned by the networks and how that relates to translation quality.

Keywords: multimodal machine translation, neural machine translation, combining language and vision, deep learning

Název práce: Multimodalita ve strojovém překladu

Autor: Jindřich Libovický

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí práce: doc. RNDr. Pavel Pecina, Ph.D.,
Ústav formální a aplikované lingvistiky

Abstrakt:

Tradičně se většina úloh zpracování přirozeného jazyka řeší výhradně uvnitř jazyka, kdy modely spoléhají na distribuční vlastnosti slov. Hluboké učení se svojí schopností učit se vhodné reprezentace vstupních dat umožňuje využití více informací tím, že trénovací signál nepochází pouze z jazyka, ale o i z obrazové modality. Jednou z úloh, které se pokoušejí využít vizuální informace, je multimodální strojový překlad: překlad popisků obrázků, kdy je stále k dispozici původní obrázek, který lze využít jako vstup pro překladač.

Tato práce shrnuje metody společného zpracovávání jazykových dat a fotografií s využitím hlubokého učení. Uvádíme přehled metod, které se využívají k řešení multimodálního strojového překladu a popisujeme náš původní příspěvek k řešení této úlohy. Představujeme metody kombinování více vstupů z potenciálně různých modalit v modelech sekvenčního učení založených na rekurentních neuronových sítích a neuronových sítí s mechanismem sebepozornosti. Uvádíme výsledky, kterých jsme dosáhli při řešení multimodálního strojového překladu a dalších úloh souvisejících se strojovým překladem. Na závěr analyzujeme, jak multimodalita ovlivňuje sémantické vlastnosti větných reprezentací, které v sítích vznikají, a jak sémantické vlastnosti reprezentací souvisí s kvalitou překladu.

Klíčová slova: multimodální strojový překlad, neuronový strojový překlad, kombinování zpracování jazyka a obrazu, hluboké učení

Acknowledgements

I thank the world for being as it is. Just because of the sheer luck of being born at the end of the 20th century in the heart of Europe, I did not have to work hard on a farm, take care of 12 younger siblings, struggle with hunger, social or political oppression or die in a battlefield of a purposeless war. Thanks to that, I was able to study long enough to write this thesis.

Many thanks belong to my colleagues with whom I had the pleasure to cooperate, especially Jindra Helcl and my supervisor Pavel Pecina, but also all other great colleagues and friends from the Institute of Formal and Applied Linguistics who make the institute an incredibly friendly place to work.

Finally, I owe special thanks to my fiancée Magda for her endless patience and support not only during the time I worked on this thesis.

The work on this thesis was supported by the Czech Science Foundation (grant number P103/12/G084) and Charles University Grant Agency (grant number 52315). This work has been using language resources and tools developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

Contents

1 Introduction

Computational Linguistics is a research field that declares the goal to invent and develop mathematical models of natural languages and propose technological applications of these models. Due to this goal, it necessarily is a multidisciplinary field which combines expertise from formal linguistics, Artificial Intelligence (AI), mathematics, and software engineering.

Attempts to automatic processing of natural languages have appeared from the early days of AI (O'Regan, 2016, p. 264). After all, the famous Turing test (Turing, 1950), which is generally considered to be a criterion of AI being really intelligent, assumes that an intelligent machine must be able to communicate in natural language. The AI took a different direction in the last 50 years—most importantly, it focused on attempts to model some of the most basic human cognitive abilities and development of intelligent agents that try to meet some objectives in an environment. Computational linguistics with its effort to model language without general AI behind the scene got established as a standalone research discipline.

In the last thirty years, computational linguistics has undergone a fascinating development during which it transformed from a relatively unknown research discipline into a field solving practical engineering task used by millions of users every day (Le and Schuster, 2016). When the focus lies more on solving language tasks from the engineering point of view, we talk rather about Natural Language Processing (NLP) than Computational Linguistics. The goal of NLP is then delivering solutions for tasks like automatic speech recognition or machine translation. Acquiring new knowledge about human language becomes secondary.

Thirty years ago, it may have seemed that automatic NLP can provide the same assurance of truth to the linguistic theories as engineering innovations provide to theories in physics. Methods of NLP used linguistic theories during the development and the source code was usually packed with explicit linguistic knowledge.

At the turn of the 21st century, it started to appear that machine-learning-based systems learning from annotated data worked better than those where the programmers embedded the linguistic knowledge explicitly into the source code. With the increasing availability of data and computational power, the amount of linguistic knowledge required to develop a solution for NLP tasks decreased. In these days, complex systems such as automatic speech recognition, machine translation or text summarization are trained from the data with no linguistics inside. The technologies found their own way, working totally independently of the language understanding provided by linguistics.

This puts us in a unique situation where the language technologies exist outside of the conceptual framework provided by classical linguistics. The situation is also different from natural science. Results of experiments in physics can often be predicted using established theories. When researchers conduct machine learning experiments in NLP, there is no theory that could in advance say what the results will be. There is only researcher's or developer's (usually strongly mathematically grounded) intuition which gets either confirmed or not.

Recent advances in deep learning, a machine learning technique using artificial neural networks, reached a new state of the art in most of the NLP and Computer Vision (CV) tasks. The deep learning models are usually trained end-to-end. Their input is data in a raw form (numerical values of pixels, sequences of words or character) without complicated preprocessing, and they produce a directly usable output. Not only end-to-end trained models perform better than methods based on explicitly programmed rules, but they also perform better than statistical systems based on carefully engineered features. It might be for the first time in the history of computer science, when we are able to develop systems that apparently know something human knowledge and conceptualization is not capable to capture yet.

This thesis follows this trend. In the following chapters, we try to push forward the state of the art in Multimodal Machine Translation (MMT) using deep learning. We believe that working on a topic that combines NLP and CV has a big potential not only to help to develop new technologies but also to contribute to the discussion of how the language relates to the extra-linguistic world. Linguistics teaches us that words in a language are signs which can be described as relations of the signifier and

the signified (Saussure, 1916). Unlike most NLP tasks, which deal explicitly only with the signifiers, tasks combining language and vision are the first real-world tasks that need to tackle also what the signs stand for not in an abstract fashion as in tasks like named entity recognition, but in concrete instances in the images.

MMT is machine translation of image descriptions from one language to another when using both the image description in the source language and the image as the input. It can also be viewed as a combination of image captioning and machine translation, two tasks in which state of the art has been recently reached using deep learning methods (Bahdanau et al., 2014; Xu et al., 2015) called *sequence-to-sequence learning*. Solving MMT requires developing methods for combing the visual and the textual input.

In this thesis, we present our novel methods for combining multiple sources in commonly used sequence-to-sequence architectures along with other techniques for improving MMT quality. We summarize our experience from three years of participation in the MMT shared task at Workshop of Machine Translation (WMT). We also present our contribution to the standard dataset (Elliott et al., 2016) for MMT, for which we created a Czech version of the dataset that was used in the 2018 WMT shared task.

In Chapter 2, we bring a comprehensive overview of the recent development of deep learning for CV and NLP. First (Section 2.2), we discuss the machine learning innovations in CV which later found use in other fields including NLP. Second (Section 2.3), we thoroughly discuss model architectures used in NLP, in particular: a transition from discrete inputs to continuous representation, architectures for sequence processing, and generating discrete outputs. In Chapter 3, we provide an overview of combining vision and language both for obtaining a grounded language representation and for solving more practically motivated tasks. Chapter 4 introduces the task of MMT and the dataset we work with, including of the Czech version of the Multi30k dataset.

Chapter 5 describes our original contribution to solving the task of MMT. In particular, we present our innovations to the deep learning architectures for sequence-to-sequence learning that allow combining multiple sources while generating a single output sequence (Sections 5.1 and 5.2). We demonstrate the contribution of our methods on MMT task and textual multi-source sequence-to-sequence learning tasks. We also provide an overview of how our methods were used by others when approaching different tasks that require processing multiple inputs. In the following part of the chapter (Section 5.3), we experiment with data augmentation and improving the system performance via multi-task learning.

Finally, in Chapter 6, we provide an analysis of the models presented in Chapter 5. First, we analyze how the quality of the system outputs depends on the objects in the image and on linguistic features of the source sentences. Later, in Section 6.2 we introduce a method for intrinsic evaluation of the representations learned by deep learning NLP models. We asses all the models presented in this thesis and draw conclusions about the representation learning abilities of the model architectures we worked with.

2

Deep Learning for Language and Vision

In this chapter, we introduce a collection of Machine Learning (ML) practices that the research and engineering community calls *deep learning*. There is no clear consensus on what should be called deep learning. No matter what exact definition we use, the term always refers to a set of practices used in ML which utilize neural networks with multiple layers and continuous optimization.

In this chapter, we discuss basic concepts of deep learning in Section 2.1, its contribution to the development of Computer Vision (CV) in Section 2.2 and Natural Language Processing (NLP) in Section 2.3.

CV is a field of computer science that deals with processing and understanding of digital images and videos (Ballard and Brown, 1982; Sonka et al., 2007). The tasks that CV addresses include object classification, object detection, face recognition, scene reconstruction, etc.

NLP is also a field of computer science but is mainly concerned with interaction between humans and machines in natural languages and ultimately understanding human language using machines (Manning and Schütze, 1999; Jurafsky and Martin, 2009). NLP includes tasks like machine translation, information retrieval, sentiment analysis or question answering. Besides the practically motivated tasks, NLP includes intermediate tasks like part-of-speech tagging or syntactic parsing, which may serve as a component in more complex pipelines, but more importantly help theoretical understanding of natural languages and can also be used for linguistic research.

The nature of the data that the two fields work with fundamentally differs. Whereas in CV, we work with high-dimensional real-valued raw data produced by sensors, in NLP we work with discrete symbols produced by humans. The important feature that these fields have in common is that they ultimately try to automate tasks which otherwise require human cognitive effort. Often, these are tasks that pose almost no difficulties for humans, but appear to be tremendously difficult to be tackled computationally.

2.1 Fundamentals of Deep Learning

As already stated, *deep learning* is a branch of ML that does not have an exact definition. It usually means ML with neural networks which have many layers (Goodfellow et al., 2016). By 'many', people usually mean more than experts before 2006 used to believe was numerically feasible (Hinton and Salakhutdinov, 2006; Bengio et al., 2007). In practice, they are networks with dozens of layers. First, it were unsupervised methods for layer-wise pre-training (Bengio et al., 2007) that demonstrated the potential of deeper neural networks. This was followed by innovations allowing training the models end-to-end by error back-propagation only (Srivastava et al., 2014; Nair and Hinton, 2010; Ioffe and Szegedy, 2015; Ba et al., 2016; He et al., 2016) without any pre-training, which allowed the boom of deep learning methods after 2014.

Neural networks and other ML models are trained to fit training data, while still being able to generalize for unseen data instances. During training, we try to minimize an error the model makes on the training data. To ensure that the model can make correct predictions on data instances that were not used for training, we use another dataset, usually called the validation set which is only used for estimating the performance of the model on unseen data.

2.1.1 Perceptron Algorithm

Deep learning originates in studying artificial neural networks (Goodfellow et al., 2016, p. 12). Artificial neural networks are inspired by a simplistic model of a biological neuron (McCulloch and Pitts, 1943; Rosenblatt, 1958; Widrow, 1960). In the model, the neuron collects information on its dendrites and based on that, it sends a signal on the axon, its single output. Formally, we say that the artificial neuron has an input, a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ of real numbers. For each input component x_i, there is a weight $w_i \in \mathbb{R}$ corresponding to the importance of the input compo-

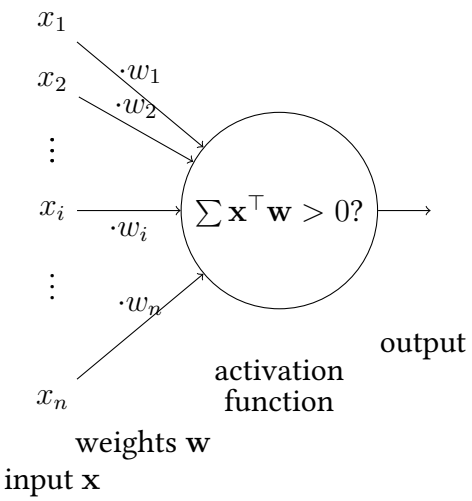


Figure 2.1: Illustration of a single artificial neuron with inputs $\mathbf{x} = (x_1, \ldots, x_n)$ and weights $\mathbf{w} = (w_1, \ldots, w_n)$.

nent. The weighted sum of the input is called the *activation*. We get the neuron *output* by applying the *activation function* on the activation. In the simplest case, the activation function is the signum function. More activation functions are discussed in Section 2.2. The model is illustrated in Figure 2.1.

The first successful experiments with such a model date back to 1950s and the geometrically motivated perceptron algorithm (Rosenblatt, 1958) for learning the model weights. The model is used for classification of the inputs into two distinct classes. The inputs are interpreted as points in a multi-dimensional vector space. The learning algorithm searches for a hyperplane separating one class of the inputs from the other. The trained weights are interpreted as a normal vector of the hyperplane. The algorithm iterates over the training examples. If an example is misclassified, it rotates the hyperplane towards the misclassified example by subtracting the input from the weight vector. It can be proved that this simple algorithm converges to a separating hyperplane if it exists (Novikoff, 1962). The linear-algebraic intuition developed for the perceptron algorithm is important also for the current neural networks.

During the following 60 years of development of ML and Artificial Intelligence (AI), neural networks fell out of the main research interest, especially during the so-called AI winters in the 1970s and 1990s (Crevier, 1993, p. 203).

In the rest of the chapter, we do not closely follow the history of neural networks but only discuss the innovations that seem to be the most important from the current perspective and relevant to our research. Techniques which are particularly useful for CV and NLP are then discussed in Sections 2.2 and 2.3, respectively. For a comprehensive overview of the history of neural network research, we refer the reader to a survey by Schmidhuber (2014).

2.1.2 Multi-Layer Networks

The geometrically motivated perceptron learning algorithm cannot be efficiently generalized to networks with a more complicated structure of interconnected neurons. In this case, we no longer interpret the learning as a geometric problem of finding a separating hyperplane. Instead, we view the network as a parameterized continuous function. The goal of the learning is to optimize the parameter values with respect to a continuous error function, usually called the *loss function*.

During training, we treat the network as a function of its parameters, given a training dataset which is considered constant at one training step. This allows computing gradients of the network parameters with respect to the loss function and updating the parameters accordingly. At inference time, the parameters are fixed and the network is treated as a function of its inputs with constant parameters.

The original perceptron used the signum function as the activation function. In order to make the function defined by the network differentiable, it was often replaced by sigmoid function or hyperbolic tangent, yielding values between -1 and 1.

For the sake of efficiency, the neurons in artificial neural networks are almost always organized in layers. This allows us to re-formulate the computation as a matrix multiplication (Fahlman and Hinton, 1987). Such layers are called *fully connected* or *dense* layers. Let $\mathbf{h}_i = (h_i^0, \ldots, h_i^n) \in \mathbb{R}^n$ be the output of the i-th layer of the network and the $(i+1)$-th layer $A : \mathbb{R} \to \mathbb{R}$ activation function. The value of the k-th neuron in the $(i+1)$-th layer of dimension m is

$$h_{i+1}^k = A\left(\sum_{l=0}^{n} h_i^l \cdot w_i^{(l,k)} + b_i^{(k)}\right) \tag{2.1}$$

which is in fact definition of matrix multiplication. It thus holds:

$$\mathbf{h}_{i+1} = A\left(\mathbf{h}_i \mathbf{W}_i + \mathbf{b}_i\right) \tag{2.2}$$

where $\mathbf{W}_i \in \mathbb{R}^{n \times m}$ is a parameter matrix and $\mathbf{b}_i \in \mathbb{R}^m$ is the bias.

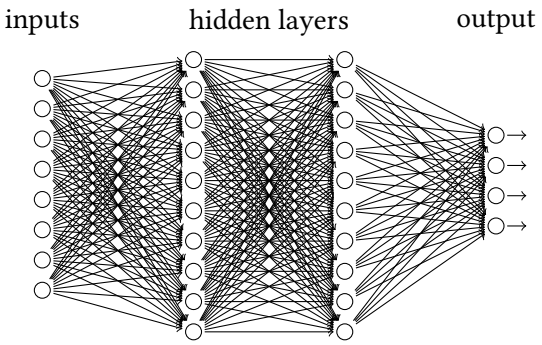


Figure 2.2: Multi-layer perceptron with two fully connected hidden layers.

Not only did this make this the computation efficient, but it also led to a re-conceptualization of the network architectures. Current literature no longer talks about single neurons, but always about network layers. This re-conceptualization then allows innovations like attention mechanism (Bahdanau et al., 2014), residual connections (He et al., 2016) or layer normalization (Ba et al., 2016) which conceptually, not only for the sake of computational efficiency, treat the neuron outputs as elements of vectors and matrices.

A network with feed-forward fully connected layers is illustrated in Figure 2.2. This architecture is usually called *multi-layer perceptron*, even though it is not trained with the perceptron algorithm but the using the error back-propagation algorithm.

2.1.3 Error Back-Propagation

With the error back-propagation algorithm, the models are trained iteratively. In every step of the model training, we compute values of the loss function, i.e., the error the network makes on the training data, or more often a small subset of training data, called a *mini-batch* (Amari, 1993). Then, we compute the gradients of the parameters with respect to the loss function using the back-propagation algorithm (Werbos, 1990) and update the parameters accordingly. The parameter updates are done using the stochastic gradient descent or its more advanced variants. For more details, we refer the reader to Goodfellow et al. (2016, pp. 286–292).

While using the back-propagation algorithm, we represent the computation as a directed acyclic graph where each node corresponds to an input, trainable parameter or an operation. This graph is called *forward computation graph.* In order to compute the derivative of a parameter with respect to the function, we build a *backward graph*

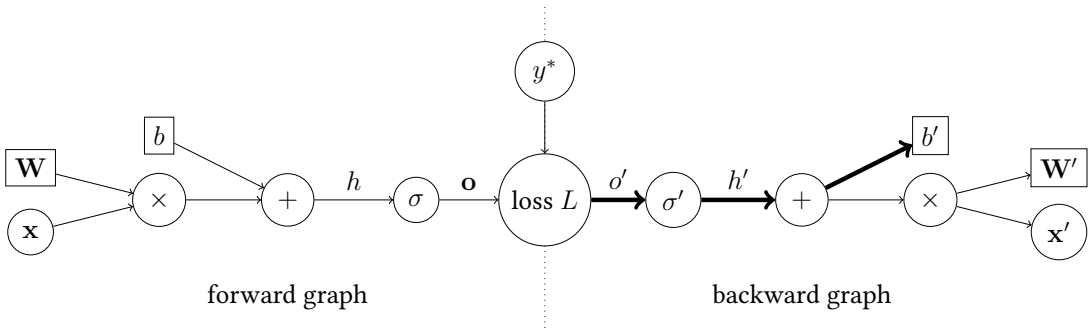


Figure 2.3: Computation graph for back-propagation algorithm for logistic regression $o = \sigma(\mathbf{W}\mathbf{x} + b)$. The highlighted path corresponds to the computation of $\frac{\partial L}{\partial b}$, which is, according to the algorithm, equal to $\frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial h} \cdot \frac{\partial h}{\partial b}$.

with reversed edges and operations replaced by their derivatives. The derivative of a parameter with respect to the loss is then computed by multiplying the values on a path from the loss to a copy of the parameter in the backward graph. The algorithm is illustrated in Figure 2.3.

The back-propagation algorithm together with techniques ensuring a smooth gradient flow within the network and regularization techniques allows training models end-to-end from a raw input. During the training process, neural networks develop an input representation such that the task becomes almost trivial to solve (Bengio et al., 2003; LeCun et al., 2015).

2.1.4 Representation Learning

Many problems both in NLP and CV can be interpreted as searching for suitable data representation. A naive representation, such as a sequence of characters or a set of image pixels, is not well-suited for further processing, although they contain full information that humans can interpret almost without effort.

Deep learning dramatically changes how data is represented. In NLP, the text used to be tokenized and enriched by automatic annotations that include part-of-speech tags, syntactic relations between words or entity detection. This representation was usually used to get meaningful features for a ML model. In statistical Machine Translation (MT), words are represented by monolingual and bilingual co-occurrence tables which are used for probability estimations within the models. In deep learning models, text is represented with tensors of continuous values which are not explicitly designed but implicitly inferred during model optimization.

This is often considered to be one of the most important properties of neural networks. Goodfellow et al. (2016, p. 5) even consider the representation learning ability to be the feature that distinguishes deep learning from the previous ML techniques. In both CV and NLP models, consecutive layers learn more contextualized and presumably more abstract representation of the input. As we will discuss in the following sections, the representations learned by the networks are often general and can often be reused for solving different tasks than they were trained for.

2.2 Deep Learning Techniques in Computer Vision

In this section, we discuss the main concepts and approaches that deep learning brought into CV. We limit this introduction to static images and focus on the image classification task.

Before the advent of deep learning, bottom-up approaches dominated CV (Sonka et al., 2007). Image understanding started from computationally inexpensive primitives like edges, color blobs, extremal regions or recognized patterns. The further processing utilized either rule-based or machine-learned combination of these building elements.

Deep learning models are usually trained in an end-to-end setup, i.e., the input of the model is an image in a raw form, and all the processing steps are trained jointly within one model. The basic deep learning tools that are used in computer vision are Convolutional Neural Networks (CNNs) usually consisting of 2D convolutional and max-pooling layers.

2.2.1 Convolutional Networks

In this section, we consider an image to be a table of three-channel (RGB: red, green, blue) pixels, i.e., a three-dimensional tensor. Note that if we disregard the exact number of channels, this is the same form as input and outputs of most of the network layers. This allows us to treat the input in the same way all other layers in the network.

In general, an input of a convolutional layer is a three-dimensional tensor $\mathbf{X} \in mathbbR^{h \times w \times c}$ of height h, width w and c channels in its third dimension. A convolution of kernel size k (typically 3 or 5) and p filters, is a non-linear projection of sub-tensors of size $k \times k \times c$ into p-dimensional vectors. Formally, the vector at position i, j in the layer output $\mathbf{H}$ is defined as

$$\mathbf{H}_{ij} = A\left([X_{i+m,j+n}]_{\substack{m=-k/2,\ldots,+k/2 \\ n=-k/2,\ldots,+k/2}} \cdot \mathbf{W} + \mathbf{b}\right) \tag{2.3}$$

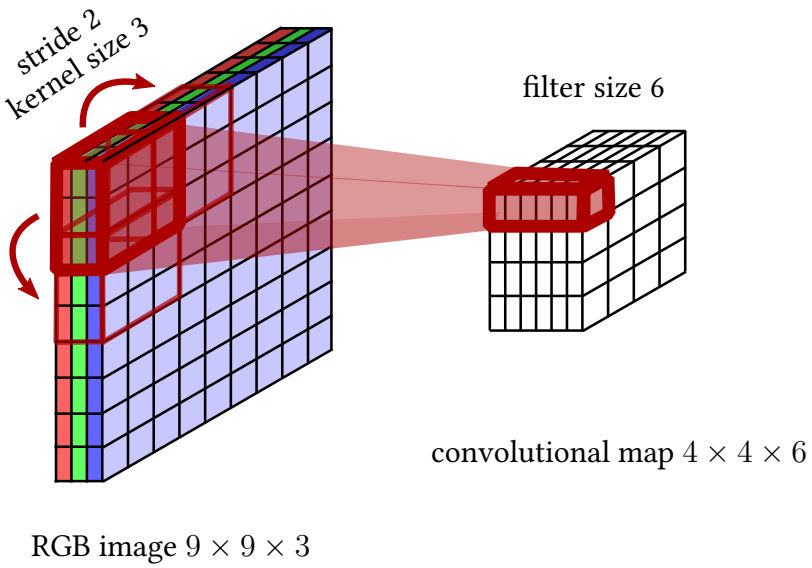


Figure 2.4: Illustration of a 2D convolution over a 9×9 RGB image with stride 2, kernel size 3 and number of filters 6.

where $\mathbf{W} \in \mathbb{R}^{c \times p}$ and $\mathbf{b} \in \mathbb{R}^p$ are trainable parameters and $A : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function.

2D convolutions can be explained as applying a sliding window projection over the input tensor which measures the similarity between the input window and filters. Another attribute of the convolution is the *stride* which is the size of the step by which the window moves. Note that the resulting feature map is always stride times smaller in the first two dimensions. A 2D convolution over an RGB image is illustrated in Figure 2.4.

Max-pooling is a dimensionality reduction technique that is used to decrease information redundancy during image processing. Analogically to the convolutions, we process sub-tensor of size $k \times k \times p$ with max-pooling, but instead of projecting it, we take the element-wise maximum in the third dimension. Formally for input tensor $\mathbf{X}$,

$$H_{i,j,k} = \max_{\substack{m=-k/2,\ldots,+k/2 \\ n=-k/2,\ldots,+k/2}} X_{i+m,j+n,k}. \tag{2.4}$$

Convolution is usually interpreted as a latent feature extraction over the input tensor where the filters correspond to the latent features. Max-pooling can be interpreted as a soft existential quantifier applied over the window, i.e., the result of max-pooling says whether and how much the latent features are present in the given region of the image.

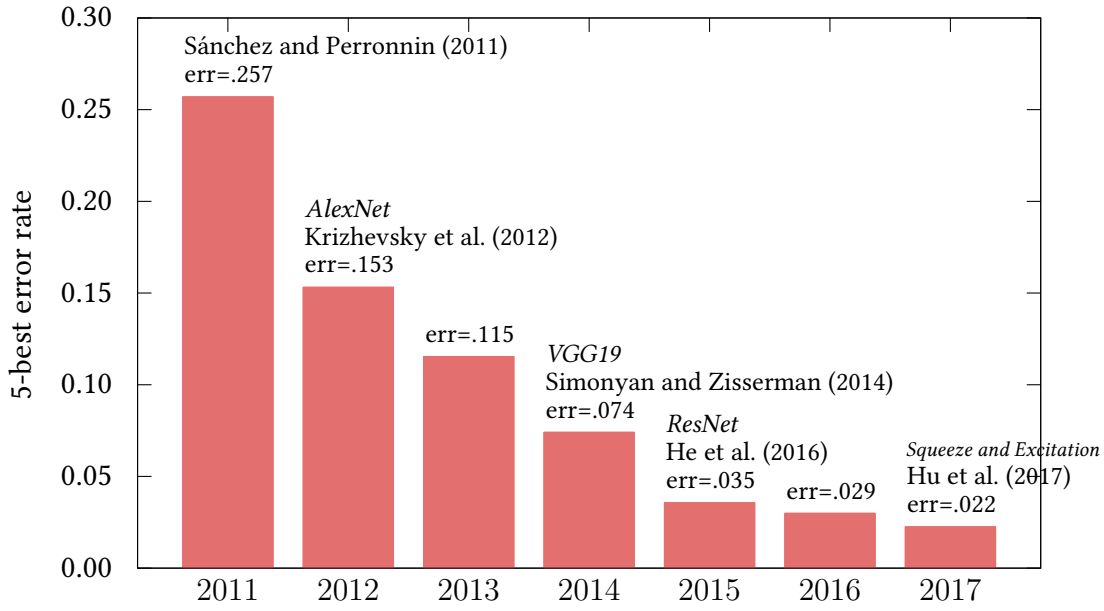


Figure 2.5: Development of performance in ImageNet image classification task between 2011 and 2017. The figures are taken from the official website of the challenge. Columns without citations correspond to submissions that did not provide a citation.

Visualizations of trained convolution filters show that the representation in the network is often similar to features used in classical CV methods such as edge detection (Erhan et al., 2009). It also appears that with the growing number of layers, more abstract representations are learned. Although in theory, shallow networks with a single hidden layer have the same capabilities (Hornik, 1991), in practice, well-trained deeper networks usually perform better (Goodfellow et al., 2016, p. 192–194).

2.2.2 Image Classification using AlexNet

The success of neural networks in CV can be well illustrated on the ImageNet challenge (Deng et al., 2009). It is an annual competition in object recognition in real-world photographs.

The challenge uses a large dataset of manually annotated images. Every image is a real-world photograph focused on one object of 1,000 classes. The classes are objects from every-day life excluding persons. The labels of the objects are manually linked with WordNet synsets (Miller, 1995). The training part of the dataset consists of 150 million labeled images, the test set contains other 150 thousand images. The standard size of the images is 225×225 pixels. The dataset is an order of magnitude bigger than all previously used datasets. Note that the word 'net' in the dataset name does not refer to neural networks but WordNet which was an inspiration for creating the ImageNet dataset.

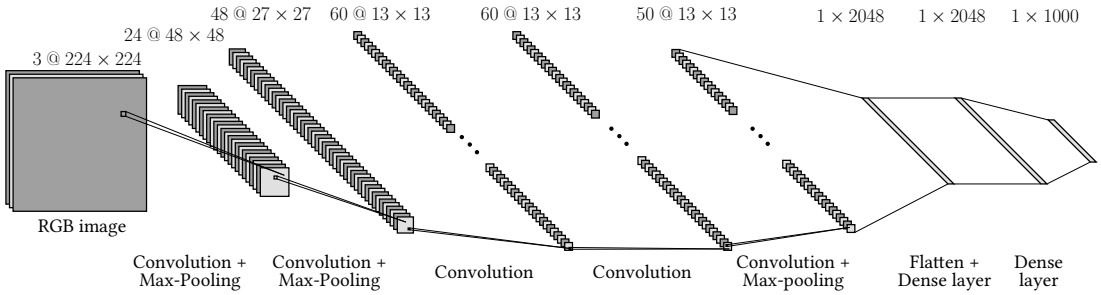


Figure 2.6: A scheme of the AlexNet architecture with stacked convolutional and max-pooling layers followed by two fully connected layers and classification layers. For clarity, we omit the technical model split for two GPUs. The visualization style is adopted from LeCun et al. (1998).

During the last 6 years, CNNs and other the deep learning techniques helped to decrease the 5-best error (proportion of cases when the correct label is not present in 5 best scoring labels), the main evaluation measure on this task more than ten times (see Figure 2.5 for more details).

CNNs were previously successfully used for simpler tasks such as handwritten digit recognition (LeCun et al., 1998). However, the first architecture that showed the potential of CNNs on large-scale tasks and made the research community focus on CNNs was AlexNet (Krizhevsky et al., 2012). The authors of this network combined many recent innovations in neural networks at the same time and developed an efficient GPU implementation, which was not common at that time. The network outperformed all previous approaches by a large margin. Moreover, the image representation learned by the network (activations in its penultimate layer) showed interesting semantic properties, allowing the network to be used to estimate image similarity based on its content.

AlexNet consists of five convolutional layers with max-pooling after the first, the second and the fifth convolutional layers and two fully connected layers of 2,048 hidden units before the classification layer, in total 7 layers with 208 million parameters. The architecture is shown in Figure 2.6.

Instead of the smooth activation functions mentioned in the previous section (2.1), it uses Rectified Linear Units (ReLUs) (Hahnloser et al., 2000; Nair and Hinton, 2010):

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.5}$$

This activation function allows better propagation of the loss gradient to deeper layers of the network by reducing the effect of the vanishing gradient problem. The derivative of hyperbolic tangent has an upper bound of one and has values close to zero on most of its domain. It makes it hardly possible to train networks with more than one or two hidden layers (AlexNet had 7 layers; Krizhevsky et al., 2012). During

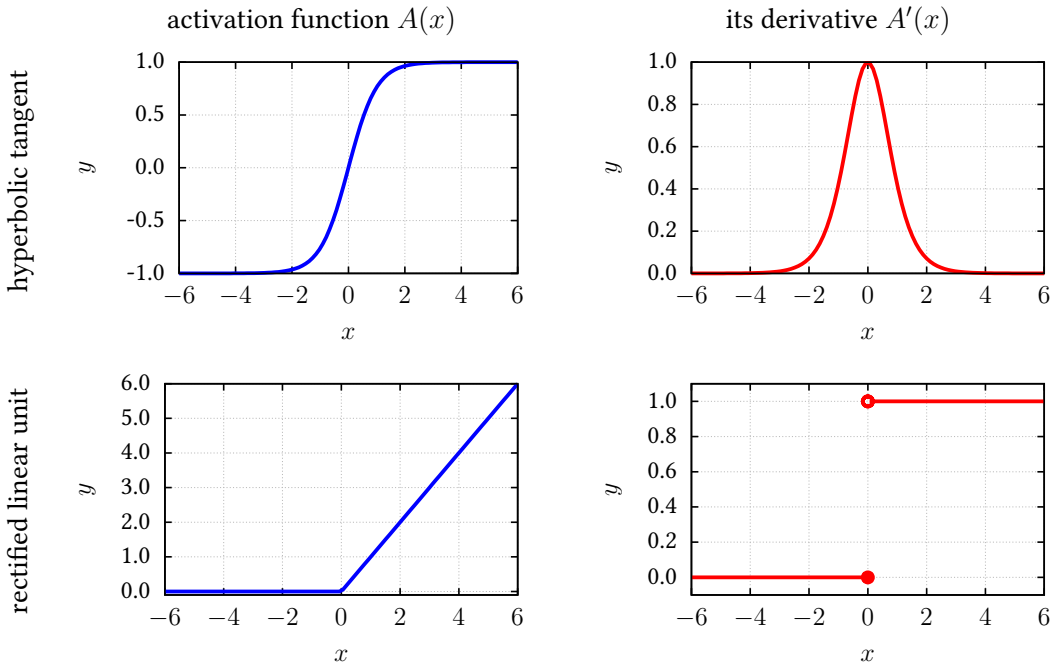


Figure 2.7: Activation functions and their derivatives.

the computation of the loss gradient with the chain rule, the gradient gets repeatedly multiplied by values smaller than one and eventually vanishes. ReLU reduces this effect, although not entirely solves this problem. However, the gradient is zero on half of the domain, which means that the probability that the gradient is zero grows exponentially with the network depth. See Figure 2.7 for visualization of courses of the activation functions and their derivatives.

The AlexNet network has 208 million parameters, which makes it prone to overfitting because it has a capacity to memorize the training set with only little generalization. AlexNet used *dropout* (Srivastava et al., 2014)[1] to reduce overfitting. It is a technique that introduces random noise in the network during training and thus forces the model to be more robust to variance in the data. With dropout, neuron outputs are randomly set to zero with a probability that is a hyperparameter of model training. In practice, dropout is implemented as multiplication by a random binary matrix after applying the activation function. Dropout can be also interpreted as ensembling of exponentially many networks with a subset of currently active neurons that share all their weights.

[1]The paper was published in a journal in 2014, however its preprint was available already in 2012 before the ImageNet competition.

2.2.3 Further Improving the Convolutional Networks

In 2014, AlexNet was significantly outperformed by the *VGG networks* (Simonyan and Zisserman, 2014) with two versions having 16 and 19 layers respectively. The authors of the network did not use any fundamentally different techniques from AlexNet. Unlike AlexNet that used convolutions with a large receptive field, VGG networks only used convolutions with kernel size 3. Using smaller kernel size reduced the number of parameters per layer and thus allowed to train a deeper network leading to a presumably more abstract image representation.

An important innovation to the network architectures was *batch normalization* (Ioffe and Szegedy, 2015). Batch normalization is a regularization technique that tries to ensure the neuron activations have zero mean and unit variance. It makes propagation of the gradient easier by keeping the neuron activations near the values where the derivatives of the activation functions vary the most.

With batch normalization, neuron activations $\mathbf{a} \in \mathbb{R}^p$ (i.e., neuron outputs before applying non-linearity, the activation function) are normalized using the sample mean $\mu \in \mathbb{R}^p$ and the standard deviation $\sigma \in \mathbb{R}^p$

$$\hat{\mathbf{a}} = \frac{\mathbf{a} - \mu}{\sigma + \epsilon} \tag{2.6}$$

where $\epsilon \in \mathbb{R}$ is a hyperparameter that prevents numerical instability if the variance is close to zero.

The mean and variance are estimated from mini-batches (tens to hundreds of examples) of training data for each neuron independently. The stochasticity of the training process with mini-batches would make the estimate numerically unstable. For this reason, we should also consider estimates from the previous training batches. However, we also need to take into account that the parameters of the network change during the training. The solution that meets these constraints is Exponentially Weighted Moving Average (Lawrance and Lewis, 1977). The current estimate is combined with the previously estimated value multiplied by a factor $0 < \alpha < 1$ which is another hyperparameter of the training.

The μ and σ estimation also requires relatively large training batches such that the mean and variance estimate is robust enough. Note also that μ and σ are adjustable parameters of the network but are trained using a different mechanism than error back-propagation.

Batch normalization allowed development of another technique which makes training of networks with many layers easier, *residual connections* (He et al., 2016). In residual networks, outputs of later layers are summed with outputs of previous layers (see Figure 2.8). Residual connections improve the flow of the gradient dur-

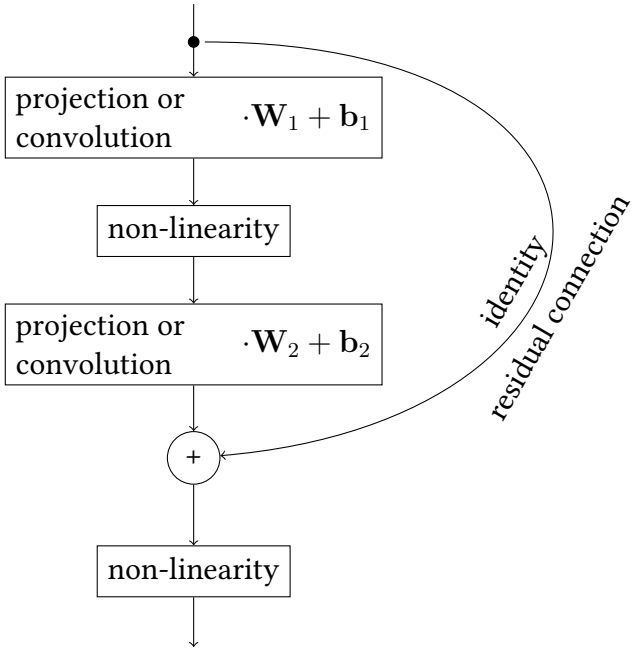


Figure 2.8: Network with residual connection skipping one layer.

ing the loss back-propagation because the loss does not need to propagate via the non-linearities causing the vanishing gradient problem. It can flow directly via the summation operator which is linear with respect to the derivative. Note also that applying the residual connection requires that the dimensionality of the layers must not change during the convolution.

Before introducing residual connections, the state-of-the-art image classification networks had around 20 layers (Simonyan and Zisserman, 2014; Szegedy et al., 2015), ResNet (He et al., 2016) used up to 150 layers while decreasing the classification error to only 3.5%.

Image classification into 1,000 classes is of course not the only task CV community attempts to solve. CV tasks include object localization (Girshick, 2015; Ren et al., 2015), face recognition (Parkhi et al., 2015; Schroff et al., 2015), traffic sign recognition (Zhu et al., 2016), scene text recognition (Jaderberg et al., 2014) and many others. Although there are many task-specific techniques, in all current approaches, images are first processed using a stack of convolutional layers with max-pooling and other techniques used also in image classification.

Representations learned by networks trained on the ImageNet dataset generalize beyond the scope of the task and seem to be aware of abstract concepts (Mahendran and Vedaldi, 2015; Zeiler and Fergus, 2014; Olah et al., 2017). The ImageNet dataset is also one of the biggest CV datasets available, often orders of magnitude bigger than

datasets for more specific tasks (Huh et al., 2016). This makes the representations learned by the image classification networks suitable to use in other CV tasks (Girshick, 2015; Branson et al., 2014; Marmanis et al., 2016) as well as tasks combining vision with other modalities (Antol et al., 2015; Vinyals et al., 2017).

2.3 Deep Learning Techniques in Natural Language Processing

Unlike the CV models where the input is always a continuous signal, in NLP, we need to deal with the fact that language is written using discrete symbols. The count and the use of the symbols, how the symbols group into words or larger units, the amount of information carried by a single symbol; this all varies dramatically across languages. Nevertheless, the symbols are always discrete. Deep learning models for NLP thus need to convert the discrete input into a continuous representation that is processed by the network before it eventually generates a discrete output.

In all NLP tasks, we can thus distinguish three phases of the computation:

- Obtaining a continuous representation of the discrete input (often called word or symbol embedding);
- Processing of the continuous representation (*encoding*) using various architectures;
- Generating discrete (or rarely continuous) output, sometimes called *decoding*.

Approaches to the phases may vary in complexity. This is most apparent in case of generating an output which can be done either using simple classification, sequence labeling techniques such as conditional random fields (Lafferty et al., 2001) or connectionist temporal classification (Graves et al., 2006) or using relatively complex autoregressive decoders (Sutskever et al., 2014).

The rest of the section discusses these three phases in more detail. First (Section 2.3.1), we discuss embedding of discrete symbols into a continuous space. In the following section (2.3.2), we discuss three main architectures that can be used for processing an embedded sequence: Recurrent Neural Networks (RNNs), CNNs and Self-Attentive Networks (SANs). The following section (2.3.3) summarizes classification and sequence labeling techniques as a means of generating discrete output. Finally, we discuss autoregressive decoding which is a technique that allows generating arbitrarily long sequences.

2.3.1 Word Embeddings

Neural networks rely on continuous mathematics. When using neural networks for NLP, we need to bridge the gap between the symbolic nature of the written language and the continuous quantities processed by neural networks. The most intuitive way of doing so is using a predefined finite indexed set of symbols called a vocabulary (those are typically words, characters or sub-word units) and represent the input as one-hot vectors. A *one-hot vector* is a vector that has zeroes everywhere except for a one at the position of the symbol that is represented by this vector. We denote a one-hot vector having one on the i-th position as $\mathbf{1}_i$. If the one-hot vector is used as input of a layer, it gets multiplied by a weight matrix. The multiplication then corresponds to selecting one column from the weight matrix. These vectors are called *symbol embeddings*.

Note also that in this setup, the only information that the networks have available about the input words is that they belong to certain classes of equivalence (usually we consider words with the same spelling to be the equivalent) indicated by the one-hot vector. The only information that the network can later work with is the co-occurrence of these classes of equivalence. The models thus heavily rely on the distributional hypothesis (Harris, 1954). The hypothesis says that the meaning of the words can be inferred from the contexts in which they are used. The success of neural networks for NLP shows that the hypothesis holds at least to some extent.

Now, consider we are going to train a neural network that predicts a probability of a word in a sentence given a window of its three predecessors, i.e., acts like a trigram Language Model (LM). The network has three input words represented by one-hot vectors with vocabulary V, and one output, a distribution over the same vocabulary. For simplicity, we further assume the network has one hidden layer $\mathbf{h} \in \mathbb{R}^m$ of dimension m before the classification layer. Formally, we can write:

$$\mathbf{h} = \tanh\left(\mathbf{1}_{w_{n-3}}\mathbf{W}_3 + \mathbf{1}_{w_{n-2}}\mathbf{W}_2 + \mathbf{1}_{w_{n-1}}\mathbf{W}_1 + \mathbf{b}_h\right) \tag{2.7}$$

$$\mathsf{P}(w_n) = \mathrm{softmax}(\mathbf{W}\mathbf{h} + \mathbf{b}) \tag{2.8}$$

where $\mathbf{W}_i \in \mathbb{R}^{|V|\times m}$ are the embedding matrices for the words in the window of predecessors and $\mathbf{W} \in \mathbb{R}^{m\times|V|}$ a projection matrix from the hidden state $\mathbf{h}$ to the output distribution, $\mathbf{b}_h$ and $\mathbf{b}$ are corresponding biases.

All four projection matrices have $|V|\cdot m$ parameters. With the vocabulary size of ten thousand words and the hidden layer with hundreds of hidden units, this means millions of parameters. All three embedding matrices have a similar function in the model. They project the one hot vectors to a common representation used in the

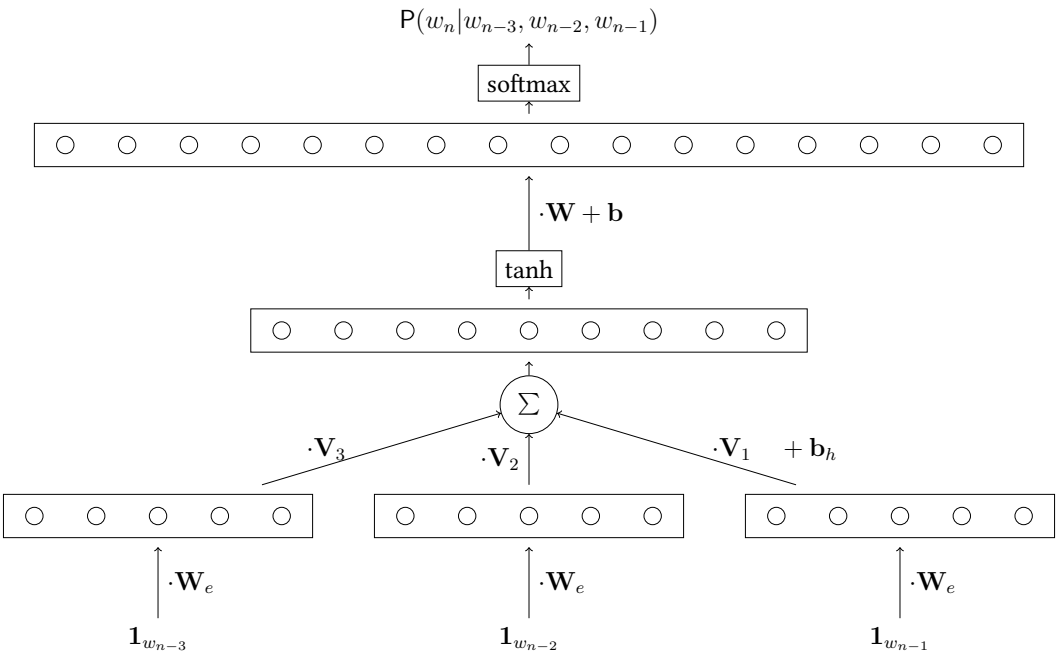


Figure 2.9: Feed-forward architecture of a language model with window size 3 with shared word embeddings $\mathbf{W}_e$.

hidden layer, also reflecting the position in the window of the predecessors. The target representation space used by the hidden layer should be the same because the output classifier cannot distinguish where the values came from unless the weight matrices learn this during model training.

Given this observation, we can factorize the matrices into two parts: the first one performing the projection to a common representation space of dimension m that can be shared among the window of predecessors, and the second projection adapting the vector to the specific role in the network based on the word position. Formally:

$$\mathbf{h} = \tanh\left(\mathbf{1}_{w_{n-3}}\mathbf{W}_e V_3 + \mathbf{1}_{w_{n-2}}\mathbf{W}_e\mathbf{V}_2 + \mathbf{1}_{w_{n-1}}\mathbf{W}_e\mathbf{V}_1 + \mathbf{b}_h\right) \tag{2.9}$$

where $\mathbf{W}_e \in \mathbb{R}^{|V|\times m}$ is the shared word embedding matrix and $\mathbf{V}_i$ are smaller projection matrices of size $m \times m$. This step approximately halves the number of network parameters. This is also the way that word embeddings are currently used in most NLP tasks. The architecture of the described trigram LM is illustrated in Figure 2.9.

The previous thoughts led us exactly to the architecture of the first successful neural LM (Bengio et al., 2003). The feed-forward architecture not only achieved decent quantitative results in terms of corpus perplexity, but it also developed word representations with interesting properties. Words with similar meaning tend to have

similar vector representations in terms of Euclidean or cosine distance. Moreover, the learned representations appear to be useful features for other NLP tasks (Collobert et al., 2011). The reasons for introducing the embedding matrix are similar also in case of RNN and CNN architectures discussed in the next section.

Mikolov et al. (2010) trained an RNN-based LM for speech recognition where the word representations manifest another interesting property. The vectors seemed to behave linearly with respect to some semantic shifts, e.g., words that differ only in gender tend have a constant difference vector. Mikolov et al. (2013) further examined this property of the word vectors and developed a simple feed-forward architecture that was no longer a good LM but still produced word embeddings with all the interesting properties, i.e., being useful machine-learning features for NLP tasks, clustering words with similar meaning and behaving linearly with respect to some semantic shifts.

Pre-trained embeddings using one of the above-mentioned methods are an important building block in NLP tasks with limited training data (dependency parsing: Chen and Manning, 2014, Straka and Straková, 2017; question answering: Seo et al., 2016) when the model is supposed to generalize for words which were not seen in the training data, but for which we have good pre-trained embeddings. In tasks with a large amount of training data such as MT, we usually train the word embeddings together with the rest of the model (Qi et al., 2018).

Development of universally usable word vector representations became an independent subfield of NLP research. The research community mostly focuses on studying theoretical properties of the embeddings (Levy and Goldberg, 2014; Agirre et al., 2016) and multilingual embeddings either with or without the use of parallel data (Luong et al., 2015a; Conneau et al., 2017).

2.3.2 Architectures for Sequence Processing

In NLP, we usually treat the text as a sequence of tokens which correspond to words, subwords, or characters. Deep learning architectures for sequence processing thus must be able to process sequential data of different lengths. The length of sentences processed by the MT systems typically varies from a few words to tens of words. In the CzEng parallel (Bojar et al., 2016b) 90% of sentences have between 20 and 350 tokens.

Currently, there are three main types of architectures used: RNNs, CNNs, and SANs. The architectures are explained in detail in the following sections.

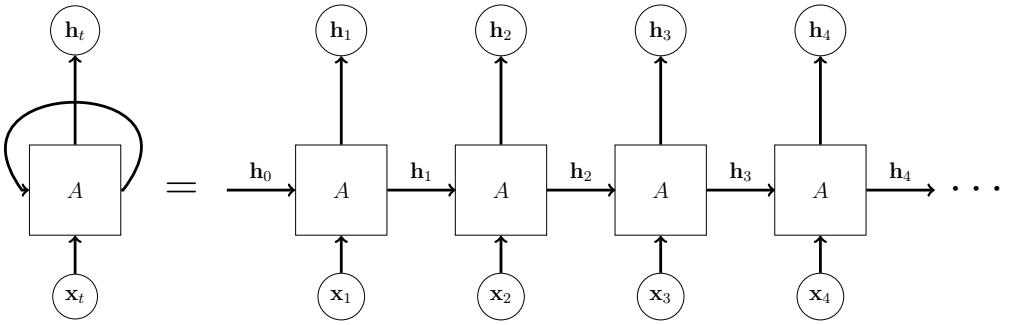


Figure 2.10: States of an RNN unrolled in time.

Recurrent Networks

RNNs are historically the oldest and probably still the most frequently used architecture for sequence processing in a variety of tasks including speech recognition (Graves et al., 2013; Chan et al., 2016), handwriting recognition (Graves and Schmidhuber, 2009; Keysers et al., 2017) or neural machine translation (Bahdanau et al., 2014; Chen et al., 2018). It was the architecture of the first choice partially because of its theoretical strengths—RNNs are proved to be Turing complete (Siegelmann and Sontag, 1995)—and because an efficient way for training them has been known since 1997 (Hochreiter and Schmidhuber, 1997).

Unlike the feed-forward networks which are stateless, a recurrent network can be best described as applying the same function A sequentially on the previous network state and current input (Elman, 1990). Computation of a new state $\mathbf{h}_t \in \mathbb{R}^d$ from the previous state $\mathbf{h}_{t-1} \in \mathbb{R}^d$ and current input $\mathbf{x}_t \in \mathbb{R}^n$ can be described using a recurrent equation

$$\mathbf{h}_t = A(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{2.10}$$

where the initial state $\mathbf{h}_0$ is either fixed or a result of previous computation. Depending on the output of the task, either the final state of the RNN $\mathbf{h}_{T_x}$ where T_x is the length of the input sequence or the whole matrix $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{T_x}) \in \mathbb{R}^{T_x \times d}$ is used for further processing.

For inference, only the current state of the network is required. However, to learn its parameters via back-propagation in time (Werbos, 1990), we need to unroll all its steps. In this sense, even a simple RNN is a deep network because the back-propagation must be conducted through many unrolled layers. From the training perspective, RNNs in NLP tasks can easily have tens or hundreds of layers. Unrolling the network is illustrated in Figure 2.10.

The depth of the unrolled network is the factor that makes training of such architectures difficult. With a simple non-linear activation function (so-called Elman cell, Elman, 1990):

$$\mathbf{h}_t = \tanh\left(\mathbf{W}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}\right), \tag{2.11}$$

it would be impossible for the network to learn to also consider longer dependencies in the sequence due to the *vanishing gradient problem* (already discussed in Section 2.2).

The derivative of the network state $\mathbf{h}_t$ in time t with respect to bias $\mathbf{b}$ from Equation 2.11 applied several time steps before t is:

$$\begin{aligned}
\frac{\partial \mathbf{h}_t}{\partial \mathbf{b}} &= \frac{\partial \tanh(\overbrace{\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}}^{=\mathbf{z}_t \text{ (activation)}})}{\partial b} \quad \text{(tanh}' \text{ is derivative of tanh)} \\
&= \tanh'(z_t) \cdot \left(\frac{\partial \mathbf{W}_h \mathbf{h}_{t-1}}{\partial \mathbf{b}} + \underbrace{\frac{\partial \mathbf{W}_x \mathbf{x}_t}{\partial \mathbf{b}}}_{=0} + \underbrace{\frac{\partial \mathbf{b}}{\partial \mathbf{b}}}_{=1} \right) \\
&= \underbrace{\mathbf{W}}_{\sim\mathcal{N}(0,1)} \underbrace{\tanh'(\mathbf{z}_t)}_{\in(0;1]} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{b}} + \tanh'(\mathbf{z}_t).
\end{aligned}$$

The derivative of $\mathbf{h}_t$ with respect to $\mathbf{b}$ gets multiplied in each step by a number between zero and one which effectively prevents the network from learning to consider also longer dependencies.

ReLU activation is claimed to reduce the issue in the context of CV (see Section 2.2). Its derivative is zero for $x < 0$ and one otherwise, so the gradient can eventually vanish in case of longer sequences.

Another type of numeric instability that can occur during RNN training is the *exploding gradient problem* (Pascanu et al., 2013). This type of instability is caused by repetitive multiplication by the same matrix during the back-propagation.

A solution to the instability problems came with introducing the mechanism of Long Short-Term Memory (LSTM) networks, which ensures that during the error back-propagation, there is always a path through which the gradient can flow via operations that are linear with respect to the derivative. The path, sometimes called information highway (Srivastava et al., 2015), is illustrated as a red straight line on the top of Figure 2.11.

This configuration is achieved by using two distinct hidden states, private state $\mathbf{C}$ and public state $\mathbf{h}$ where the state $\mathbf{C}$ is updated using the linear operations only. A gating mechanism explicitly decides what information from the input can enter the information highway (*input gate*), which part of the state should be deleted (*forget gate*) and what part of the private hidden state should be published (*output gate*).

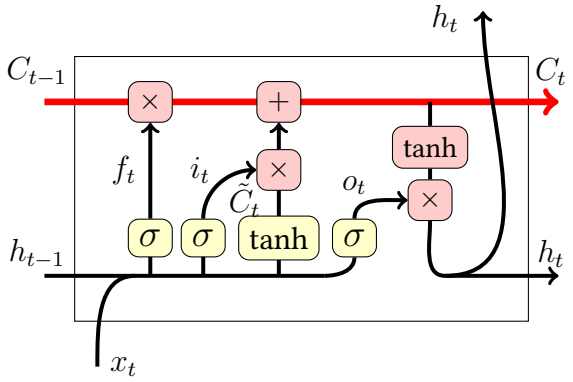


Figure 2.11: A scheme of an LSTM cell with the information highway on the top of the scheme. Non-linear projections are in yellow boxes, point-wise operations in pink boxes, variables denoted at the arrows correspond to Equations 2.12 to 2.17.

Formally, LSTM network of dimension d updates its two hidden states $\mathbf{h}_{t-1} \in \mathbb{R}^d$ and $\mathbf{C}_{t-1} \in \mathbb{R}^d$ based on the input $\mathbf{x}_t$ in time step t in the following way:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_f\right) \tag{2.12}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_i\right) \tag{2.13}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_o\right) \tag{2.14}$$

$$\tilde{\mathbf{C}}_t = \tanh\left(\mathbf{W}_c \cdot [h_{t-1}; \mathbf{x}_t] + \mathbf{b}_C\right) \tag{2.15}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + i_t \odot \tilde{\mathbf{C}}_t \tag{2.16}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh \mathbf{C}_t. \tag{2.17}$$

where $\odot$ denotes point-wise multiplication. The cell is shown in Figure 2.11.

The values of the forget gate $\mathbf{f}_t \in (0,1)^d$ control how much information is kept in the memory cell by point-wise multiplication. In the next step, we compute the candidate state $\tilde{\mathbf{C}} \in \mathbb{R}^d$ in the same way as the new state is computed in the Elman RNN cells. Values of this candidate state are not combined directly with the memory. First, they are weighted using the input gate $\mathbf{i}_t \in (0,1)^d$ and added to the memory already pruned by the forget gate. The new output state $\mathbf{h}_t$ is computed by applying $\tanh$ non-linearity on the memory state $\mathbf{C}_t$ and weighting it by the output gate $\mathbf{o}_t \in (0,1)^d$.

As previously mentioned, LSTM networks have two separate states $\mathbf{C}_t$ and $\mathbf{h}_t$. The private hidden state $\mathbf{C}_t$ is only updated using addition and point-wise multiplication. The $\tanh$ non-linearity is only applied while computing the output state $\mathbf{h}_t$. The gradient from the output passes through only one non-linearity before entering the information highway.

Later, other numerically stable versions of RNNs appeared. They all have the property that there is a path on which the gradient can propagate without vanishing (Balduzzi and Ghifary, 2016; Lee et al., 2017b). The most frequently used variant are Gated Recurrent Units (GRUs) (Cho et al., 2014b):

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_z\right) \tag{2.18}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_r\right) \tag{2.19}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}[\mathbf{r}_t \odot \mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}\right) \tag{2.20}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t. \tag{2.21}$$

The GRU networks have fewer parameters than LSTM networks which may speed up training under some circumstances. Performance of both network types is comparable and is task dependent (Chung et al., 2014).

A commonly used method for improving the RNN performance is building a bidirectional network (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). Two independent RNN networks are used in parallel, each of them processing the sequence from one end. The output states are then concatenated. In this way, the network can better capture dependencies in both directions in the input sequence. Bidirectional RNNs became a standard in many NLP tasks (Bahdanau et al., 2014; Ling et al., 2015; Seo et al., 2016; Kiperwasser and Goldberg, 2016; Lample et al., 2016). Note that in this setup, every network state may contain information about the complete sequence.

Convolutional Networks

CNNs were already discussed in detail in Section 2.2 in the context of CV. Whereas in the case of the CV tasks, convolutions are explained as applying a sliding window over an image in two dimensions, in NLP, convolutions are usually one-dimensional. CNNs applied on a sequence of states or embeddings can be interpreted as applying a sliding window over the sequence that extracts useful features of the input n-grams it processes.

Their most significant advantage over RNNs is that while processing a sequence, the computation does not need to wait for the result of processing the previous inputs and the computation can proceed in parallel. CNNs also resemble methods utilizing n-gram statistics, frequently used in older NLP techniques (Cavnar and Trenkle, 1994). Due to these properties, there have been attempts to use CNNs in NLP tasks since the advent of deep learning techniques (Kalchbrenner and Blunsom, 2013; Kim, 2014), however with only limited success.

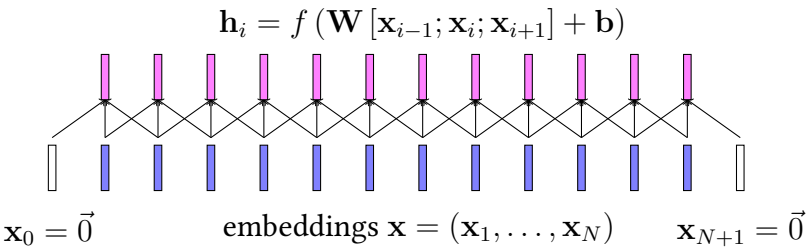


Figure 2.12: One layer of a 1-D convolution.

In the bidirectional setups, RNNs cover the whole input sequence in every output state, whereas the receptive field of the CNNs is limited to a small sliding window, in practice of up to five tokens. In NLP, it took until 2017 (Wu et al., 2017; Gehring et al., 2017) when CNNs gained popularity after techniques allowing stacking multiple layers were available (Yu and Koltun, 2015; He et al., 2016; Ba et al., 2016). They allowed enlarging the receptive field of the networks so that they succeed in more complex tasks.

A vector on j-th position in the i-th layer of a 1D-CNN window of size k (kernel size) is computed as

$$\mathbf{h}_j^{(i)} = f(W[\mathbf{h}_{j-\frac{k}{2}}^{(i-1)}; \mathbf{h}_{j-\frac{k}{2}+1}^{(i-1)}; \ldots; \mathbf{h}_{j+\frac{k}{2}}^{(i-1)}] + \mathbf{b}). \tag{2.22}$$

Applying the convolution window can be thus intuitively interpreted as extracting features from k-grams of input symbols. One layer of a CNN is illustrated in Figure 2.12.

A shallow convolution can capture only properties of n-grams without long-range dependencies and disregards mutual position of the n-grams. Single-layer CNNs are therefore suitable just for tasks that are solvable without considering broader context such as sentiment analysis (Kim, 2014) where even simple n-gram-based statistical methods yield interesting results (Pak and Paroubek, 2010).

Enlarging the size of the convolution window would soon increase the number of parameters prohibitively. A better way for expanding the receptive field of CNNs is stacking multiple layers on each other (see Figure 2.13). With multiple layers, the problem of vanishing gradients comes into play again.

This problem is approached similarly as in deep CV networks or LSTMs by modifying the architecture in such a way that the computation graph contains a path on which the gradient can flow through operations which are linear with respect to the derivative. The path is created using residual connections, which sum up the output

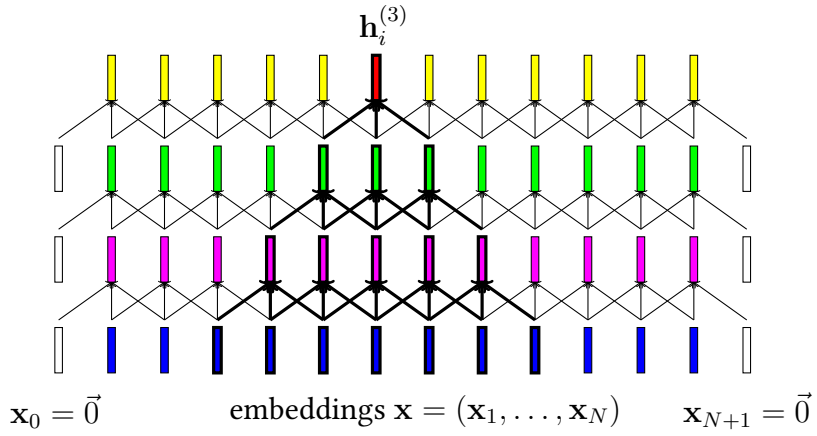


Figure 2.13: Receptive field of a multi-layer CNN. With kernel size 3, stride 1 and 3 layers, a vector in the third layer covers 7 input embedding vectors.

$$\mathbf{h}_i = f\left(\mathbf{W}\left[\mathbf{x}_{i-1}; \mathbf{x}_i; \mathbf{x}_{i+1}\right] + \mathbf{b}\right) + \mathbf{x}_i$$

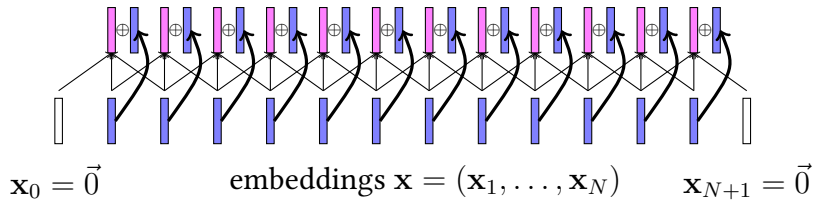


Figure 2.14: Convolutional layer with residual connections.

of the layer with its input (see Figure 2.14):

$$\mathbf{h}_j^{(i)} = f\left(\mathbf{W}[\mathbf{h}_{j-\frac{k}{2}}^{(i-1)}; \mathbf{h}_{j-\frac{k}{2}+1}^{(i-1)}; \ldots; \mathbf{h}_{j+\frac{k}{2}}^{(i-1)}] + \mathbf{b}\right) + \mathbf{h}_j^{(i-1)}. \tag{2.23}$$

As was already discussed in Section 2.2, training a network with residual connections suffers from numerical instability because the output and input of the network can shift to a different scale and one term in the summation can outweigh the other one. To stabilize the training, the output activation of the network needs to be normalized. Unlike in CV, where batch normalization (Ioffe and Szegedy, 2015) is used to address this problem, in NLP, layer normalization (Ba et al., 2016) is prevails due to its empirically better performance even with small training batches.

The activations (the values before applying a non-linear activation function) on the i-th layer are normalized:

$$\overline{\mathbf{a}}^{(i)} = \frac{\mathbf{h}^{(i)}}{\sigma^i}\left(\mathbf{a}^{(i)} - \mu^{(i)}\right) \tag{2.24}$$

where $\mathbf{g}^{(i)} \in \mathbb{R}^{d_i}$ is a layer specific trainable parameter. The scalar sample mean $\mu^{(i)}$ and deviance $\sigma^{(i)}$ are estimated as follows:

$$\mu^{(i)} = \frac{1}{d_i} \sum_{j=1}^{d_i} \mathbf{a}_j^{(i)}, \quad \sigma^{(i)} = \sqrt{\frac{1}{d_i} \sum_{j=1}^{d_i} \left(\mathbf{a}_j^{(i)} - \mu^{(i)}\right)^2}. \tag{2.25}$$

where d_i is dimension of the i-th layer.

In 2016, CNNs started to be used for obtaining character-level representations which were later processed using RNNs (e.g., character-level MT, Lee et al., 2017a; question answering, Seo et al., 2016). Later in 2017, methods using multi-layer CNNs with residual connections appeared (e.g., MT, Gehring et al., 2017; question answering, Wu et al., 2017) offering significant speedup in training while maintaining the performance of previous RNN-based models.

Unlike RNNs, CNNs are by design not aware of mutual positions of symbols in the sequence and treat all n-grams in the sequence equally. In tasks where the n-gram order plays an important role such as in MT, this can be solved by introducing position embeddings which can encode either absolute (Gehring et al., 2017) or relative position of the symbols (Shaw et al., 2018). Such embeddings are trained jointly with the model.

Although CNNs are in theory weaker models than RNNs which are known to be Turing complete, they are computationally more efficient due to the possibility to parallelize their computation. They can achieve the same results as RNNs when correctly designed and trained (Gehring et al., 2017; Wu et al., 2017).

Self-Attentive Networks

SANs are neural networks where at least for some layers, the states of the next layer $\mathbf{H}^{(i)} = (\mathbf{h}_0^{(i)}, \ldots, \mathbf{h}_T^{(i)}) \in \mathbb{R}^{T \times d}$ for a sequence of length T and dimension d are computed as a linear combination of the states on previous layer $\mathbf{H}^{(i-1)}$. Formally,

$$\mathbf{h}_k^{(i)} = \sum_j \alpha_{j,k}^{(i)} \mathbf{h}_j^{(i-1)} \tag{2.26}$$

where $\alpha_{:,k}^{(i)} \in (0; 1]^T$ is a trained probability distribution which depends on the values on layer $\mathbf{H}^{(i-1)}$.

There exist several variants of SANs (Parikh et al., 2016; Lin et al., 2017). In this section, we discuss in detail the encoder part of the architecture introduced by Vaswani et al. (2017), called *Transformer*, that achieves state-of-the-art results in MT and that we also use in our experiments with Multimodal Machine Translation (MMT).

A *Transformer layer* for sequence encoding consists of *two sub-layers*[2]. The first sub-layer is self-attentive, the second one is a non-linear projection to a larger dimension followed by a linear projection back to the original dimension. All sub-layers contain dropout, layer normalization and are connected using residual connections. The scheme of the architecture is displayed in Figure 2.16.

The self-attentive sub-layer first computes a similarity between all states using the scaled dot-product attention. Attention is usually interpreted as probabilistic retrieving of values $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathbb{R}^{n \times d}$ which are associated with some keys $\mathbf{K} = (\mathbf{k}_1, \ldots, \mathbf{k}_n) \in \mathbb{R}^{n \times d}$ for each of m query vectors $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_m) \in \mathbb{R}^{m \times d}$. We define the scaled dot-product attention as

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \tag{2.27}$$

where d is the model dimension, i.e., the second dimension of all three matrices involved in the equation, which is the Transformer model constant across the layers.

By normalizing the similarity over the keys, we get a probability distribution which is then applied over the value matrix $\mathbf{V}$ in a weighted sum as shown in Equation 2.26. In case of the self-attentive encoder, all three matrices $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are the same, i.e., the states on the next layer $\mathbf{H^{(i)}} = (\mathbf{h}_1^i, \ldots, \mathbf{h}_n^{(i)})$ are computed as:

$$\mathbf{H}^{(i)} = \sum \mathrm{softmax}\left(\frac{\mathbf{H}^{(i-1)}\mathbf{H}^{(i-1)\top}}{\sqrt{d}}\right)\mathbf{H}^{(i-1)}. \tag{2.28}$$

The dot-product had been used as a similarity measure in the attention mechanism previously (Luong et al., 2015b), Vaswani et al. (2017) added the scaling factor by $\sqrt{d}$ to prevent absolute values of the similarity to grow with the growing model dimension. The dot product is a computationally cheaper alternative to using a single-layer network (Bahdanau et al., 2014) later called feed-forward attention (Luong et al., 2015b).

Vaswani et al. (2017) also introduced another innovation to the attention mechanism, *multi-headed attention*. This technique allows collecting different information from different states of the previous layer into a single context vector. In the multi-head setup, all the query, key and value matrices are first linearly projected as illustrated in Figure 2.15. Note that even if the inputs to the attention are the same, the projections do not share parameters. The projected states are then split into multiple sub-matrices, so-called heads. The attention is computed for each of the heads

[2]Note that even the sub-layer consists of several network layers. A better term would probably be *block* as in ResNet (He et al., 2016), however, we follow the terminology introduced by Vaswani et al. (2017).

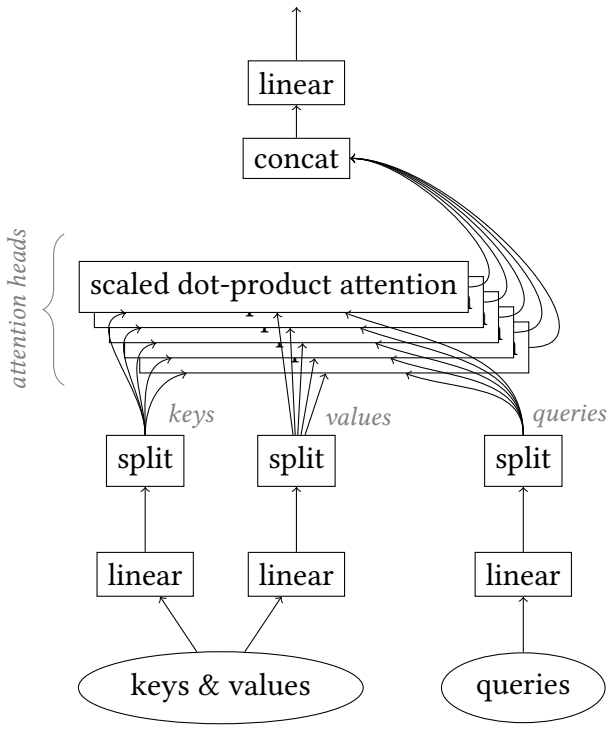


Figure 2.15: A scheme of the multi-headed scaled dot-product attention.

independently according to Equation 2.27. Outputs of the attention are then concatenated and linearly projected to the original model dimension, forming a sequence of context vectors. In the Transformer model, the keys and values are always the same. In case the attention is used as self-attention, the queries are identical as well.

The multi-head setup uses two independent projections for keys and values. The keys and values are thus different in the individual heads. Formally for h heads,

$$\begin{aligned} \mathrm{Multihead}(\mathbf{Q}, \mathbf{V}) &= (\mathbf{H}_1 \oplus \cdots \oplus \mathbf{H}_h)\mathbf{W}^O & (2.29) \\ \mathbf{H}_i &= \mathrm{Attn}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{V}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) & (2.30) \end{aligned}$$

where $\mathbf{W}^O \in \mathbb{R}^{hd \times d}$, and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times d}$ are trainable parameters.

Similar to CNNs, SANs do not have any means to capture the order of the symbols and thus require using additional positional encoding to distinguish the positions within the sequence. Instead of trained *positional embeddings* (Gehring et al., 2017), the Transformer model uses analytically computed *positional encoding* of dimension d:

$$\mathrm{pos}(i) = \begin{cases} \sin\left(\frac{t}{10^4}^{\frac{i}{d}}\right), & \text{if } i \mod 2 = 0 \\ \cos\left(\frac{t}{10^4}^{\frac{i-1}{d}}\right), & \text{otherwise} \end{cases} \quad (2.31)$$

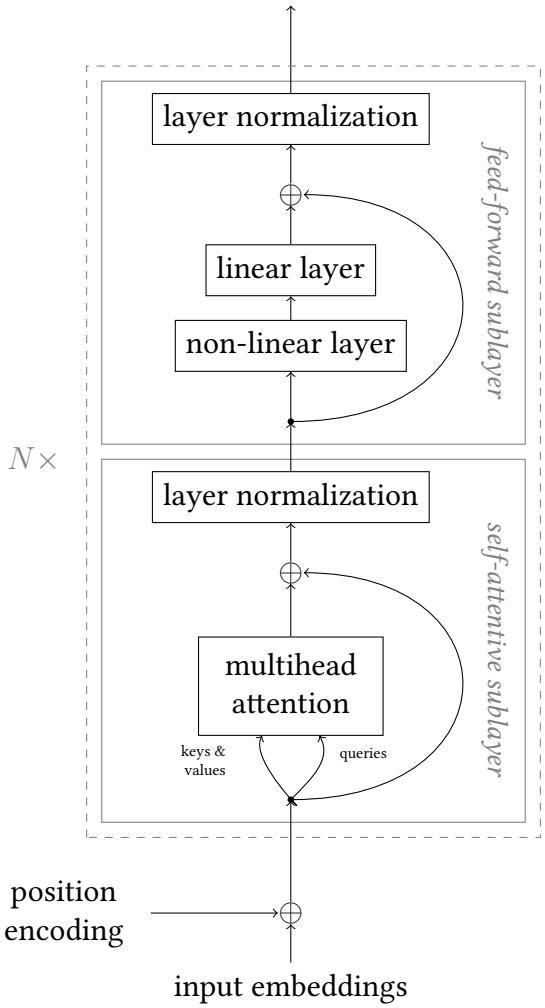


Figure 2.16: A scheme of a self-attentive encoder network from the Transformer model with N layers.

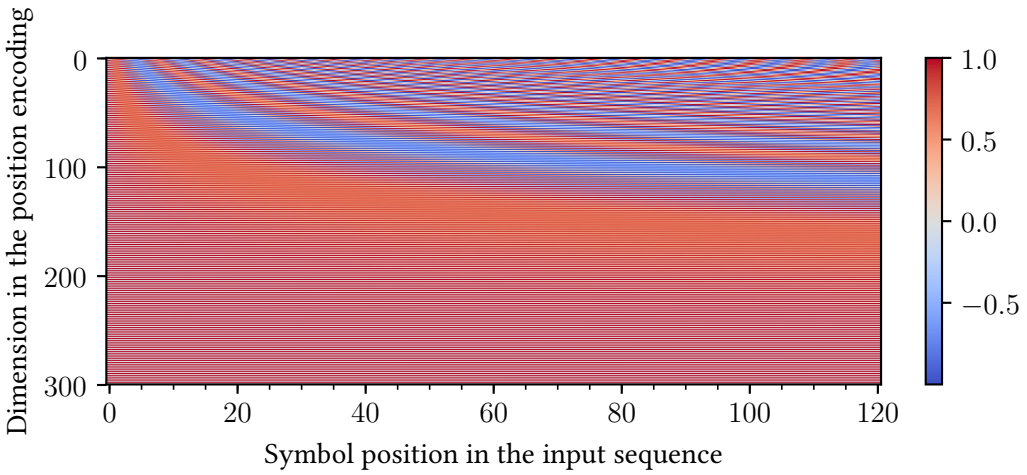


Figure 2.17: Visualization of the position encoding used in the Transformer model with embedding dimension 300 and input length up to 120.

	computation	sequential operations	memory
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n \cdot d)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(n \cdot d)$
Self-attentive	$O(n^2 \cdot d)$	$O(1)$	$O(n^2 \cdot d)$

Table 2.1: Comparison of the asymptotic computational, sequential and memory complexities of the architectures processing a sequence of length n, and state dimensionality d. CNN has kernel size k. The first column contains complexity in case of sequential computation, the second column shows the asymptotic number of sequential operation during parallel computation, and the third column shows the memory complexity.

The positional encoding for model dimension 300 is plotted in Figure 2.17. Note that for different dimensions, the encoding values change with a different frequency which allows estimating relative position of the inputs.

Computationally, SANs are as fast as convolutions because the self-attention can be computed in a single matrix multiplication which can be highly parallelized when computed on GPU. On the other hand, we need to store a matrix with similarity of all pairs of the input states in the GPU memory for each layer of the network. Due to this, the memory demands grow quadratically with the length of the input sequence. A summary of the computational complexity of the discussed networks for sequence processing is given in Table 2.1.

2.3.3 Generating Output

So far, we have only discussed how the neural networks process symbolic input into an intermediate representation. In the following sections, we discuss what architectures are used to generate an output from neural networks.

We discuss in detail three special cases:

- The output is exactly one symbol from a closed set of possible answers—*classification*;
- The output is a sequence of symbols of the same length as the input—*sequence labeling*;
- The output is a sequence of symbols of an arbitrary length—*autoregressive decoding*.

Classification

In the simplest case, the network produces only one discrete output, i.e., we want to classify the input into a fixed set of previously known classes. An example of such an NLP task is sentiment analysis (Pang et al., 2002; Pak and Paroubek, 2010) where the goal is to classify whether a text carries a positive or a negative sentiment. Another example can be classification of text into a set of genres (Kessler et al., 1997; Lee and Myaeng, 2002).

The most common approach to these tasks is applying a multi-layer perceptron over a fixed-size representation of the input. The fixed-size vector can be for instance the final state of an RNN or a result of a pooling function applied over states produced by one of the architectures mentioned in Section 2.3.2. The most frequently used methods are max-pooling and mean-pooling, computing the maximum or average of the states in time, respectively.

The classification network can then consist of multiple non-linear layers before the actual classification, which is usually done by taking the maximum or sampling from a distribution estimated by the softmax function.

The softmax function over a vector $\mathbf{l}$ is defined as:

$$\operatorname{softmax}(\mathbf{l})_i = \frac{\exp l_i}{\sum_{l_j \in \mathbf{l}} \exp l_j}. \tag{2.32}$$

Note that the softmax function is monotonic, so if we are interested only in the best-scoring prediction, at inference time, we can take the maximum value of vector $\mathbf{l}$ before the softmax function. The values of $\mathbf{l}$ are often called *logits*.

When the output of the network is estimated using the softmax function, we can measure the error that the network makes as a cross entropy between the estimated probability distribution P_y and the true output distribution, given such distribution exists. In practice, the true distribution is unknown. However, we usually assume that the true distribution exists and assigns all the probability mass to the target value y^* in the training data. This assumption might be problematic, e.g., when estimating the probability of the next word in a text. It is in fact never the case that there is only one possible follow-up word. Nevertheless, the assumption simplifies the computation of cross-entropy loss which can be then expressed as

$$L(\mathsf{P}_y, y^*) = -\log \mathsf{P}_y(y^*). \tag{2.33}$$

In this way, the derivative of the loss function with respect to the logits is

$$\begin{aligned} \frac{\partial L(\mathsf{P}_y, y^*)}{\partial \mathbf{l}} &= -\frac{\partial}{\partial \mathbf{l}} \log \frac{\exp l_i}{\sum_{l_j \in \mathbf{l}} \exp l_j} = \frac{\partial}{\partial \mathbf{l}} \left(\log \sum_{l_j \in \mathbf{l}} \exp l_j - l_{y^*} \right) = \\ &= \frac{\sum \mathbf{1}_{y^*} \exp l}{\sum \exp l} - \mathbf{1}_{y^*} = \mathsf{P}_y - \mathbf{1}_{y^*} \end{aligned} \tag{2.34}$$

where $\mathbf{1}_{y^*}$ is a one-hot vector for value y^*.

The loss gradient with respect to the logits is back-propagated to the network using the chain rule. The softmax function with cross-entropy loss is used not only in case of single-output classification but also in sequence labeling and autoregressive decoding discussed in the following sections.

For completeness, we should also mention that when the output of the network is supposed to be a continuous value, we can perform a linear regression over the input representation and optimize the estimation using a mean squared error

$$L(y, y^*) = (y - y^*)^2, \tag{2.35}$$

which is differentiable and thus the error can be back-propagated to the network.

Sequence Labeling

When the desired output of a network is a sequence of discrete symbols having the same length as the input and monotonically aligned with the input, we can apply a multi-layer perceptron over each state of the network. In this case, the labels assigned to every state are conditionally independent given the network states. The loss function used to train the network is a sum of cross entropy over the network output distributions.

In NLP, many tasks can be formulated as sequence labeling. Besides the more theoretically motivated tasks such as part-of-speech tagging or semantic role labeling, we can mention information extraction where the goal is marking entities in a text or named entity recognition.

The labels assigned to the input symbols often have their own, usually simple grammar rules. When we label beginnings and ends of sequences, we need to make sure the end symbol never comes before the start symbol. In these cases, conditional random fields can be applied over the state sequence (Lafferty et al., 2001; Do and Artieres, 2010).

If there are fewer output symbols than the states of the network, we can use a technique called connectionist temporal classification (Graves et al., 2006). This is often the case in speech recognition or handwriting recognition where the network states correspond to relatively short input signal segments. Connectionist temporal classification introduces a special output symbol for "no output" and thus is able to generate shorter sequences than the number of the input state. Note that this approach still assumes monotonic alignment of the labels with the input sequence.

Autoregressive Decoding

In some tasks such as MT or abstractive text summarization, the output cannot be monotonically aligned with the input and the number of output symbols differs from the number of the input symbols. In such cases, we need a mechanism that is able to generate output symbols in a general while loop and is conditioned on the entire input. Such a while loop is sometimes called *autoregressive decoder* because the computation in every time step depends on the previous state of the loop and previous outputs and generates the output symbols left-to-right.

Historically, autoregressive decoding has developed from discriminative language modeling using RNNs. We will thus first explain this principle on the RNN LMs and later generalize the principle for CNNs and SANs.

LMs are probabilistic models estimating the probability of sentences in a language represented by a corpus that the model is trained on. The probability is factorized over the words or smaller units. Within the statistical paradigm, word probabilities were usually estimated based on a finite window of previous words using n-gram statistics computed on a training corpus (Manning and Schütze, 1999). When approached as a sequence labeling problem, it can be done using an RNN which can, in theory, handle unlimited history (Mikolov et al., 2010; Sundermeyer et al., 2012). In both cases, the probability of a sentence is estimated using the chain rule:

$$\mathsf{P}(w_1, \ldots, w_n) = \mathsf{P}(w_1|\texttt{<s>}) \cdot \cdots \cdot \mathsf{P}(w_n|w_{n-1}, \ldots, w_1, \texttt{<s>}) \tag{2.36}$$

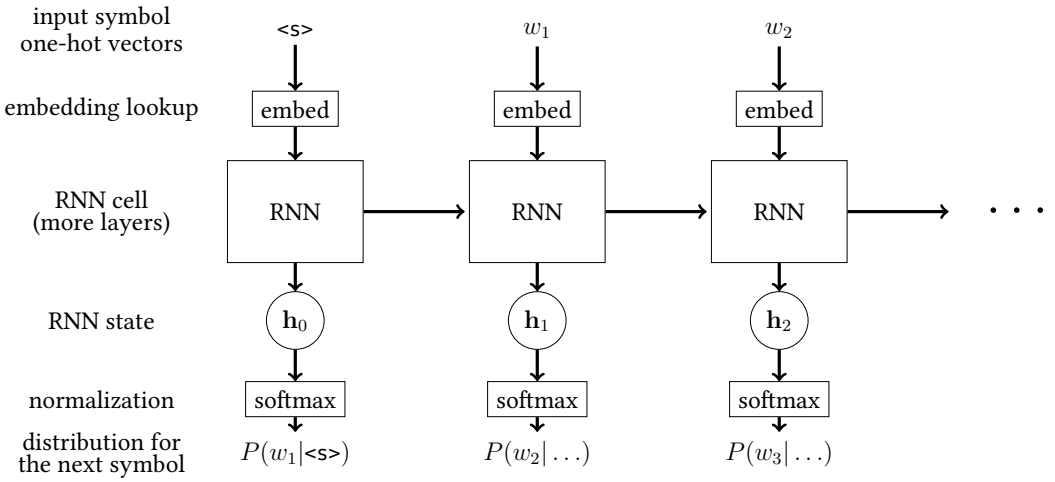


Figure 2.18: An illustration of LM formulated as a sequence labeling problem.

The inputs to the neural LM are word embeddings. In every time step, the model estimates a probability distribution over the vocabulary. Formally we let

$$\mathsf{P}(w_{n+1}|w_n, \ldots, w_1, \texttt{<s>}), \mathbf{s}_n = \mathrm{RNN}(w_n, \mathbf{s}_{n-1}) \tag{2.37}$$

where $\mathbf{s}_n$ is the state of the model in the n-th step. The distribution expresses how likely the following word w_{n+1} is to appear in a sentence with a prefix of words $w_1, \ldots, w_n$. The model is optimized towards cross entropy as a standard sequence labeling task. An RNN LM formulated as sequence labeling is illustrated in Figure 2.18.

Autoregressive decoding is based on sampling from such a model. In every time step, we can sample from a distribution over the output words. In the next step, the sampled word is provided as the model input as if it were a word in a sentence that the LM is supposed to score. The words are sampled from the model in a while loop until a special end symbol is generated. The sampling is illustrated in Figure 2.19.

The missing part that distinguishes a LM from an autoregressive decoder is conditioning the LM on other inputs than the previously decoded symbols. In case of MT, it would be the source sentence. In the simplest case, this can be done by explicitly assigning the initial state of the RNN with a result of a previous computation.

In the early work, these were max-pooled CNN states (Kalchbrenner and Blunsom, 2013), however, more promising results are achieved with LSTM networks (Sutskever et al., 2014). A disadvantage of this approach is that the input needs to be represented as a fixed-sized vector (max-pooled CNN states, the final state of an RNN) regardless of what the input length is. Performance of such models quickly decreases with the size of the input (Sutskever et al., 2014).

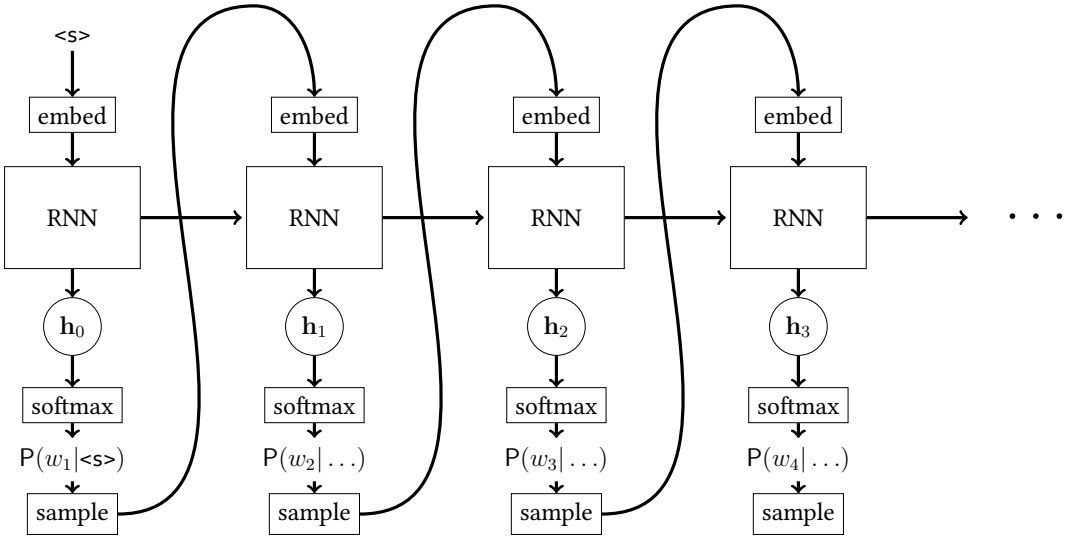


Figure 2.19: RNN LM used as an autoregressive decoder.

Bahdanau et al. (2014) introduced a technique overcoming this drawback of autoregressive decoding, called *attention mechanism*. In every decoding step, the model computes a distribution over the variable-length input representation and uses it to compute the *context vector*, a weighted average over the input representation. Having been originally introduced in the context of Neural Turing Machines (Graves et al., 2014), the distribution is usually interpreted as addressing the input representation analogically to addressing memory cells in a random-access memory.

For input sequence of length T_x, encoder states $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_{T_x}) \in \mathbb{R}^{T_x \times d_n}$ of dimension d_h, and decoder state $\mathbf{s}_i$ of dimension d_s, the attention model with intermediate dimension d_a defines the attention energies $e_{ij} \in \mathbb{R}$, attention distribution $\alpha_i \in \mathbb{R}^{T_x}$, and the context vector $\mathbf{c}_i \in \mathbb{R}^{d_h}$ in the i-th decoder step as:

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_i + \mathbf{U}_a \mathbf{h}_j + \mathbf{b}_a) + b_e, \tag{2.38}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{2.39}$$

$$\mathbf{c}_i = \sum_{j=1}^{T_x} \alpha_{ij} \mathbf{h}_j. \tag{2.40}$$

The trainable parameters $\mathbf{W}_a \in \mathbb{R}^{d_s \times d_a}$ and $\mathbf{U}_a \in \mathbb{R}^{d_h \times d_a}$ are projection matrices that transform the decoder and encoder states $\mathbf{s}_i$ and $\mathbf{h}_j$ into a common vector space and $\mathbf{v}_a \in \mathbb{R}^{d_a}$ is a weight vector over the dimensions of this space, $\mathbf{b}_a \in \mathbb{R}^{d_a}$ and $b_e \in \mathbb{R}$ are biases for the respective projections. The context vector $\mathbf{c}_i$ is then concatenated with the decoder state $\mathbf{s}_i$ and used for classification of the following output symbol.

Together with techniques for data preprocessing (Sennrich et al., 2016a,b), the attention model was the crucial innovation that helped to improve neural MT quality over the statistical MT and set a new state of the art in the field of MT.

As already mentioned, autoregressive decoding is not limited only to RNNs. The same principle can be applied to CNNs. A convolutional decoder (Gehring et al., 2017) applies a stack of 1D convolutional layers on the sequence generated so far and uses its last, i.e., the right-most state to compute a distribution over the possible output symbols from which we can sample and append the sample to the already generated sequence. Note that in each step, all states of the CNN was already computed in the previous step of the decoder, except for the last one which can be added in constant time. The decoder utilizes the attention mechanism in the same way as the RNN decoders. Gehring et al. (2017) use deep CNN of 15 layers both in the encoder and the decoder and uses the attention mechanism between the corresponding layers.

The autoregressive nature of decoding prevents parallelization of the decoder computation at inference time. At training time, the target sentence is known in advance and all the intermediate states of the convolutional decoder can be computed in parallel. Convolutional sequence-to-sequence models offer a significant training time speedup while maintaining similar performance as recurrent models (Gehring et al., 2017).

SANs can be used similarly to CNNs. In the Transformer model (Vaswani et al., 2017), a stack of self-attentive and feed-forward sub-layers is applied on the already decoded sequence and the result is used to produce the next output symbol. Similar to the CNN decoder, the self-attentive layers are interleaved with cross-attentive layers attending the encoder states (see Figure 2.20).

At training time, the computation can be parallelized as in the case of the CNN decoder. In the self-attentive layers, we need to limit the attention distribution only to the words that have already been decoded. This is in practice implemented by multiplying the matrix of attention energies with a triangle matrix as shown in Figure 2.21.

Self-attentive sequence-to-sequence models currently provide the best results in sequence generation tasks (Bojar et al., 2018). Nevertheless, they suffer from the low decoding speed and quadratic memory demands which limit practical application to sequences of at most hundreds of tokens.

So far, we have only discussed how the output sequence probability is modeled using the autoregressive decoder. Finding the sequence that receives the highest probability by the decoder is however a difficult problem. The number of possible output sequences grows exponentially with its length which makes an exhaustive search intractable.

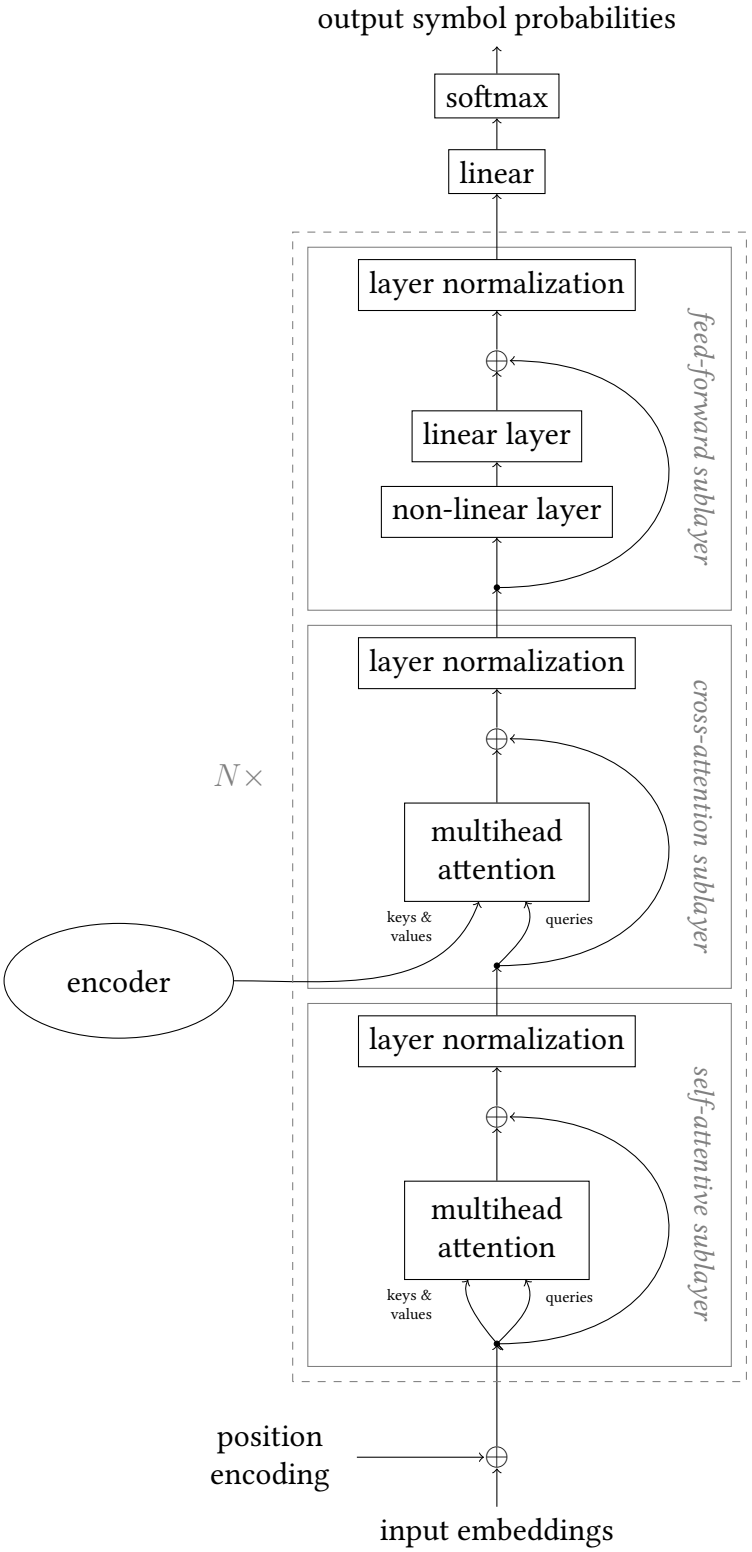


Figure 2.20: A scheme of a self-attentive decoder network from the Transformer model with N layers.

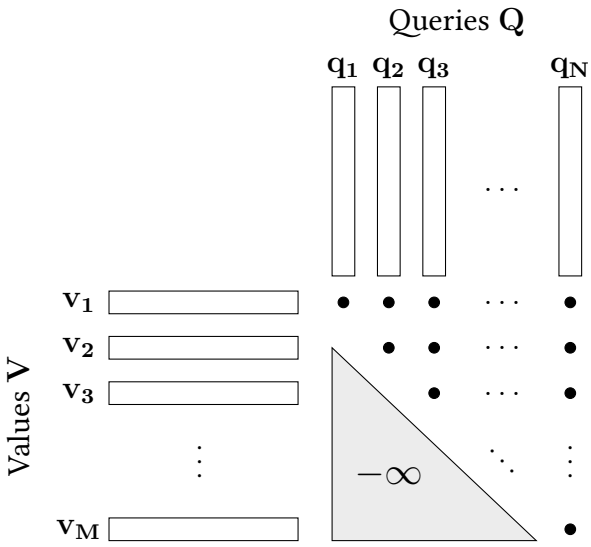


Figure 2.21: Masking while computing energy values in the self-attention layer in the Transformer decoder. Masking prevents the self-attention to attend to symbols to the right from the previously decoded symbol.

The most straightforward heuristics is greedily choosing the most probable output symbol in every step. Another commonly used heuristic is the *beam search* algorithm that simulates the exhaustive search while keeping only a few best-scoring hypotheses in every time step (Sutskever et al., 2014). The algorithm that trades off the efficiency of greedy decoding (when the output symbol with maximum probability is selected at each time step) with maintaining a relatively wide search space.

The algorithm keeps track of k hypotheses (*beam*). A hypothesis is either a partially generated sequence (*unfinished*), or a sequence that ends with a special end symbol (*finished*). In each step, all hypotheses are expanded with all possible tokens from the vocabulary. The expanded hypotheses are scored and k best of them are kept for the next step. The beam search algorithm is illustrated in Figure 2.22.

In the basic variant of the beam search decoding, the score of a hypothesis is computed using the chain rule (see Equation 2.36). The scores computed in this way do not allow comparing scores of hypotheses of different lengths. The probability score of a sequence assigned by the chain rule decreases exponentially with the sequence length. This problem is particularly apparent in models with large output vocabularies and high perplexities of the output distribution.

A frequently used feature of the scoring function is *length normalization* that tackles this issue. Length normalization divides the log sequence probability by the sequence length. The final score can thus be interpreted as the geometric mean of token probabilities. Nematus (Sennrich et al., 2017), a toolkit for neural sequence-to-sequence learning divides the sequence log probability by t^α where t is the length

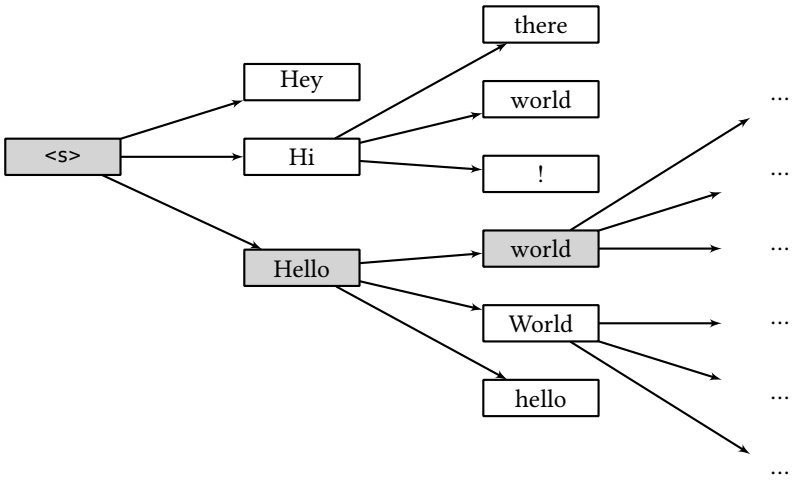


Figure 2.22: Beginning of beam search generating sentence "Hello world!" with a beam of width 2. In every step, each active hypothesis is expanded by three possible continuations from which two best scoring (marked with gray background) are selected for the next step.

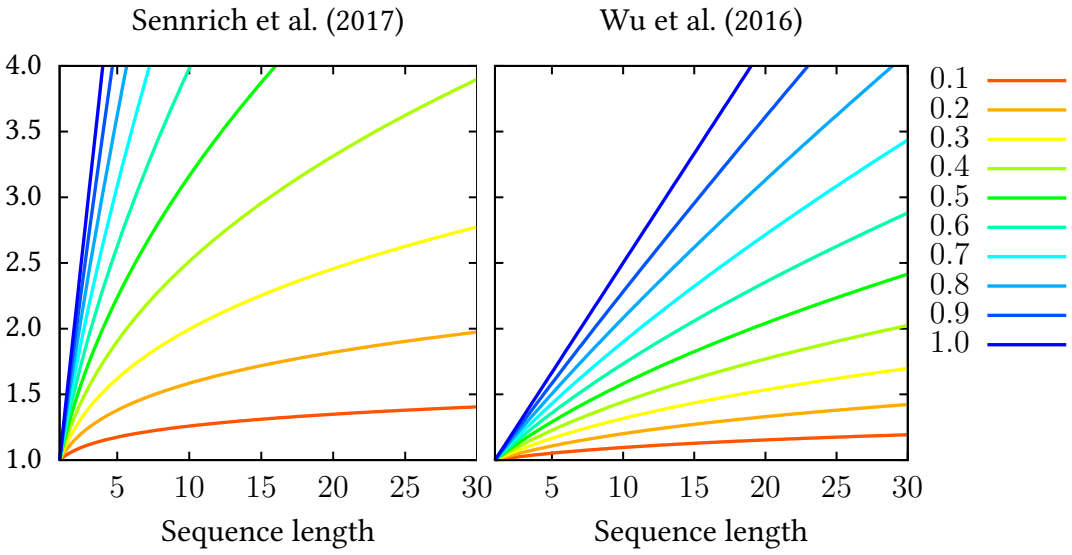


Figure 2.23: Length normalization term for beam search according to Sennrich et al. (2017) and Wu et al. (2016) with different values of hyperparameter α.

of a partial hypothesis being decoded. Wu et al. (2016), in a paper that claims to describe a production system by Google, use a heuristic formula

$$\frac{(5+t)^{\alpha}}{(5+1)^{\alpha}}. \tag{2.41}$$

In both cases, t denotes the step of the decoder (i.e., the maximum length of a partially decoded hypothesis) and α is a hyperparameter. This is the heuristic that we use in most of our experiments. The length normalization terms are visualized in Figure 2.23.

3

Combining Language and Vision

Computational linguistics has always been concerned with ambiguous sentences like: "I saw a man with a telescope" which are particularly challenging for morphological, syntactic and semantic analysis in most of the theoretical frameworks. The truth is that most of such sentences are no longer ambiguous if we consider broader context in which the sentences are spoken or written. In many cases, it is the visual context that can be used to resolve the ambiguities.

Deep learning methods which are used similarly both in Computer Vision (CV) and Natural Language Processing (NLP) allow combining visual and textual information seamlessly. It is the first time in the history of computer science when the methodology established in both research communities is almost the same. The ability of deep learning models to learn a general and information-rich representation of input data allows learning representations within a single modality and reuse it in a multimodal context, or the other way round, train a more robust multimodal representation and use it for single-modality tasks.

The rest of this section discusses more deeply how the modalities can be combined (Section 3.1), and presents an overview of the most important tasks combining language and vision which are important for our work on Multimodal Machine Translation (MMT): multimodal representation learning (Section 3.2) and image captioning (Section 3.3).

3.1 From Language Models to Multimodal Models

In NLP, language modeling was for a long time considered to be one of the most important tasks (Manning and Schütze, 1999, p. 5). Statistical Language Models (LMs) estimate a probability that a sentence can be produced in a given language and are usually trained on large corpora of text. Language models can be used as tools for distinguishing which sequence of words forms a better utterance in a given language. If we followed Chomsky's terminology, we can say that we model *linguistic competence* inferred from corpora of *linguistic performances*. The important difference between a probabilistic LM and modeling language by writing rules based on language syntax is that the LMs are trained on sentences that were actually spoken or written with some purpose, and they relate to the extra-linguistic reality. Because of this implicit relation to the world, LMs necessarily carry some semantic and pragmatic information as well. LMs do not only model syntactic correctness of a sentence, but also in some sense capture how the sentence relates to the world.

For most NLP applications, it is a great benefit. It actually means that LMs are also implicitly models of the world without the necessity to step outside the language. Moreover, texts in corpora that we use for training our models usually do not expect that its author shares much of the context with the reader. They indeed share cultural context and some texts may rely on intertextuality. Nevertheless, the author and the reader are rarely at the same place at the same time, they are not in the same weather, they do not share what they currently see. Therefore, the language context itself must be informative.

In the context of deep learning, modeling a language also includes generation continuous vector representations of words and sentences. This creates new possibilities on how to evaluate the quality of LMs. A good model of a language should not only assign a high probability to plausible sentences but also yield representations that capture important aspects of the language—and thus could be reused in other NLP tasks. For the same reasons as mentioned above, the representation must also carry some world knowledge.

Traditionally, language models are trained using language data only. Even though they can capture semantic or even pragmatic information, they cannot capture in any sense how the words or sentences refer to extra-linguistic phenomena, simply because they are not present during the model training. If we want to obtain LMs that at least somehow model language denotation, we need to train them multimodally.

Representations learned by neural networks for CV and NLP tasks manifest many similarities. Most importantly, similar inputs get similar representation even in cases when resolving the similarity requires non-trivial semantic reasoning (Deselaers and Ferrari, 2011; Mikolov et al., 2013). This naturally raises a research question of whether the representations can be somehow projected between each other and used for solving tasks that integrate language and vision.

Apart from the research questions regarding continuous representations, more engineering oriented tasks appeared. These are most importantly automatic image captioning, that we thoroughly describe in the following chapter, and visual question answering (Antol et al., 2015; Zhang et al., 2016; Goyal et al., 2017). The goal of visual question answering is to select a correct single-word answer to a natural language question about a photograph. An example of a challenging task combining language and vision that still awaits a satisfactory solution is generating images to textual descriptions (Reed et al., 2016). Grounding language in the visual modality can go even beyond the written text. Harwath and Glass (2017) used a pre-trained image classification network and spoken descriptions of images to learn word-like units from the audio and relate them using the attention mechanism with areas in the images.

3.2 Visually Grounded Representation

Acquiring visually grounded language representation is not only an intriguing research challenge, but also one of the main assumptions that MMT might bring any improvement over the standard text-only Machine Translation (MT), or help in other NLP tasks. The notion of grounding can be, in theory, seen both ways between language and vision. The annotation that is used for training CV models (e.g., object recognition or detection) is in fact already grounded in language because it uses labels which are already part of some language conceptualization of the world. When talking about representation grounding, we do not mean this implicit grounding of CV models in language, but the opposite direction: grounding language representation explicitly via having access to a visual representation of what language denotes.

Multimodal representations are often evaluated by how efficiently they can be used to retrieve objects from one modality given an object from other modality as a query. Another way of the representation quality evaluation is measuring the correlation of the representation distances with the semantic similarity of words or sentences as assessed by human annotators.

For words embeddings, Silberer and Lapata (2014) introduced a dataset for evaluation of semantic and visual similarity. Their evaluation is based on an observation that some concepts that are semantically similar might not be visually similar and vice versa. If a model succeeds in grounding the representation in the visual modality, the learned word embeddings should not only capture the semantic similarity that usually emerges while training a LM, but also the visual similarity.

Multimodal sentence representations are evaluated as retrieval models on image captioning datasets, most frequently Flickr30k (Young et al., 2014) and MS COCO (Chen et al., 2015). See Section 3.3 for more details on the datasets.

Representations can also be evaluated using downstream tasks. Large collections of such tasks for sentence representation evaluation are available (Conneau and Kiela, 2018; Wang et al., 2018a). Using standardized methods for representation evaluation requires getting representations of millions of sentences and training models for dozens of tasks using the representation which makes the evaluation often inconveniently slow.

The first experiments with visual grounding of neural language representation focused on word embeddings. Silberer and Lapata (2014) used autoencoders on word properties extracted automatically from a corpus and image vectors from a pre-trained Convolutional Neural Network (CNN) to get a grounded representation. Their evaluation shows that word embeddings trained in this setup capture both the semantic similarity between words and the visual similarity.

A more elegant method for grounding word embeddings in visual modality was introduced by Lazaridou et al. (2015) who extended the skip-gram model (Mikolov et al., 2013). During training, the model simultaneously predicts what words are in the neighborhood of a given word and a representation of an image that shows a corresponding concept. The multimodally trained models showed superior performance when evaluated on capturing semantic similarity between words.

Chrupała et al. (2015) trained visually informed sentence representation with a model called Imaginet. The model is a bidirectional Recurrent Neural Network (RNN) trained using a multi-task learning objective. The RNN was simultaneously trained in a LM setup and as a predictor of the image representation.

A body of work that does not aim to develop a general multimodal representation focuses on image retrieval based on a textual description. These models are usually trained using objectives based on Canonical Correlation Analysis (CCA) (Hardoon et al., 2004; Andrew et al., 2013; Benton et al., 2017). CCA finds a linear projection that maximizes correlation between two collections of data. Whereas the standard CCA has a closed-form solution, the generalized versions back-propagate the correlation into a neural network and update the weights as in case of any other loss function.

The common representation can then be learned with a bag of words model (Gong et al., 2014) or a randomly initialized RNN (Yan and Mikolajczyk, 2015) and a pre-trained CNN for an image. An alternative to CCA-based methods for image retrieval are methods using a ranking-based objective (Wang et al., 2017).

3.3 Image Captioning

A Language and Vision task that attracts a lot of attention is automatic image caption generation. The goal of the task is to generate a sentence in natural language which is a description of a photograph provided as an input to the model. The task is usually approached with supervised learning.

The generated captions are automatically evaluated using standard MT metrics such as BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2011). The number of possible correct descriptions of an image is enormous and the variability of output that can be considered correct is much higher than in MT. Unlike MT evaluation, where we usually use a single reference sentence, image captioning is usually evaluated on four or five independently manually created image captions. More details on the evaluation metrics are provided in Section 4.1.2.

There are several standard datasets which are used for model training and evaluation. The most frequently used are:

- *Flickr8k* (Rashtchian et al., 2010) contains crowd-sourced descriptions of 9,000 images randomly sampled from Flickr images published with permissive licenses. Every image is accompanied by 4 or 5 independently created captions.

- *Flickr30k* (Young et al., 2014) is a dataset of 30 thousand randomly chosen images from Flickr filtered in such a way that it should not contain images which are almost identical. Each image is accompanied by 5 crowd-sourced captions. Later, object annotations were added (Plummer et al., 2015) which are linked with words in the captions. The dataset is the basis of the Multi30k dataset (Elliott et al., 2016) that we use in our experiments with MMT.

- *MS COCO* (Lin et al., 2014; Chen et al., 2015) is a large CV dataset that contains 417 thousand photographs of common objects in a common environment. The images were crawled from Flickr in such a way that they contain 80 common objects in a common environment. Originally, the images were annotated with objects and semantic relations between the objects. Later, five crowd-sourced English captions were added to each image.

A group of people wearing snowshoes, and dressed for winter hiking, is standing in front of a building that looks like it's made of blocks of ice.

The people are quietly listening while the story of the ice cabin was explained to them.

A group of people standing in front of an igloo.

Several students waiting outside an igloo.

Figure 3.1: An example of an image and human captions form the Flick30k dataset.

In all the datasets, an image caption is a single sentence, usually in the present tense. A common feature of all the datasets is that the captions rarely contain named entities. The captions usually refer to people by their gender or activity they are involved in. An example is shown in Figure 3.1. More detailed linguistic analysis of English sentences from the Flickr30k dataset is provided in Section 4.2.

Image captioning was traditionally approached using complex pipelines combining statistical models for object detection, resolving relations between the objects and rules for generating the descriptions (Farhadi et al., 2010; Li et al., 2011; Kulkarni et al., 2011). The focus of the work lied in the CV components, the language generation usually relied on templates and rarely employed language modeling (Mitchell et al., 2012).

The paradigm shift in image captioning came with the work of Vinyals et al. (2015) who used a pre-trained CNN originally trained for image classification and an autoregressive RNN decoder for generating the captions. The model outperformed all the previous work by a large margin in the BLEU-1 metric, which was a standard at that time. Unlike the standard BLEU score, BLEU-1 considers unigram precision only. All the following work uses the n-gram precision up to 4-grams as when evaluating machine translation.

Vinyals et al. (2015) used the penultimate layer of a batch-normalized deep CNN for image classification (Ioffe and Szegedy, 2015) for extracting visual features. A projection of the image representation vector is used as an initial state of an RNN decoder (see Section 2.3.3 for more details on autoregressive decoding).

Xu et al. (2015) extended this model with the attention mechanism from neural MT models. In this captioning model, a convolutional map from the last convolutional layer is used instead of a single image vector. The convolutional map is treated as an unordered set of vectors, the same way as the encoder states in sequence-to-sequence architectures (Bahdanau et al., 2014).

Model	BLEU
CNN → vanilla RNN decoder (Vinyals et al., 2015)	31.4
CNN → attentive RNN decoder (Xu et al., 2015)	32.6
CNN → SAN decoder (Zhu et al., 2018)	33.3

Table 3.1: Performance of the image captioning models on the MS COCO dataset when using ResNeXt (Xie et al., 2017) network for image representation as reported by Zhu et al. (2018).

The attentive model by Xu et al. (2015) not only brings a quantitative improvement over the non-attentive RNN decoder, but it also makes the model more interpretable. By visualizing the convolutional maps over the original image, we can estimate what part of the image was used for generating particular words in the output.

Lu et al. (2017) extended the attentive model by introducing another gate into the Long Short-Term Memory (LSTM) cell, *sentinel gate*, which enables the decoder to ignore the visual input and predict the next word based on its previous state only. This approach even increases the interpretability of the model because it allows to explicitly determine when the decision of the decoder is based on the image and when it is based on the language context. We discuss this approach also in Section 5.1 in the context of MMT.

CNN maps can be used as input also in the Self-Attentive Network (SAN) architectures. The captioning model using self-attentive Transformer architecture with multi-headed scaled dot-product attention reaches a minor improvement over the RNN model (Zhu et al., 2018). Before using the image features in the cross-attention sub-layer, the convolutional maps are projected to a larger dimension and projected to the decoder dimension with the Rectified Linear Unit (ReLU) non-linearity. We use the same approach in the experiments with Transformer models for MMT in Section 5.2.

Performance of the captioning systems strongly depends on the quality of the underlying image representation. Xu et al. (2015) already reported that using VGG networks (Simonyan and Zisserman, 2014) brings a performance boost over AlexNet (Krizhevsky et al., 2012). Zhu et al. (2018) replicated the original experiments of Vinyals et al. (2015) and Xu et al. (2015) with ResNeXt architecture (Xie et al., 2017) and report 3.7 BLEU points improvement for the non-attentive RNN model and 8.3 BLEU points improvement for the attentive RNN model. The comparison of the models with the same image representation is displayed in Table 3.1.

The above-mentioned results suggest that image representation may carry rich semantic information. Madhyastha et al. (2018) showed that this interpretation might not be as straightforward as the authors of the original papers claimed. Madhyastha et al. (2018) trained a captioning model that used binary features of objects detected in the image (Redmon et al., 2016), and a model which used as an input a low-dimensional projection of the image representation. The caption quality was almost identical to models utilizing the image representation from ResNet (He et al., 2016). Their conclusions are that the image representation is not likely to be semantically richer than indication vectors of detected objects from the 80 MS COCO classes. Moreover, the fact the low-dimensional projection of the ResNet image representations is as informative as the vectors themselves suggests that the models rely on relatively shallow image similarity than on presumably deep semantic features present in the representation from ResNet.

4

Multimodal Machine Translation

In this chapter, we introduce Multimodal Machine Translation (MMT), currently one of the most intensively studied problems that combine language and vision. It is defined as translation of image captions when having both the caption in the source language and the image as the system input. The advances in the task were annually evaluated within a shared task at the Workshop of Machine Translation (WMT) workshop (Specia et al., 2016; Elliott et al., 2017; Barrault et al., 2018). Since introducing the task in 2015 (Elliott et al., 2015), the translation quality of MMT systems almost doubled in terms of BLEU. The task allows evaluating under what circumstances visual information might be useful and identifies the main challenges in combining two different modalities.

The structure of the chapter is as follows: First, we introduce the task and describe theoretical and practical motivation (Section 4.1). Then, we introduce the standard dataset used for training and evaluation MMT models (Section 4.2) including its Czech version created by us. Finally, Section 4.3 provides an overview of the state-of-the-art MMT models.

4.1 Task Definition and Motivation

In this section, we present a definition and description of Machine Translation (MT) and MMT, discuss the motivation behind introducing the MMT task and identify the main issues connected with solving the task.

4.1.1 Machine Translation

In general terms, MT can be defined as translation from one (source) natural language into another (target) natural language by using machines only (Dorr et al., 1998; Lopez, 2008). The drawback of a straightforward definition like this is that in order to evaluate how successful an MT system is, we would need to find an exact definition of what translation is. A simple claim that translation is a process of generating a text in the target language that has the same meaning as the text in the source language is not helpful as well. Deciding whether two texts have the same meaning is a tremendously difficult task and attempts to solve it would only delve us deeper into more complex theoretical questions.

The MT community thus opted to approach MT as a behaviorist simulation. People are capable of translating from one language into another and judge what a good translation is even if they are not able to rigorously define what translation is. This idea bases both how MT systems are developed and how they are evaluated. MT systems are trained on examples of translations produced by humans. Automatic MT evaluation then measures similarity of the system outputs to manually created reference sentences (Hovy et al., 2002; Papineni et al., 2002). In fact, they measure how well the system simulates an expert human behavior.

4.1.2 MT Evaluation

As discussed previously, MT is approached as behaviorist simulation. Because it is difficult to formulate any external objective criteria what translation quality is, we only measure how much the system outputs resemble translation produced by humans. This is a difficult task as well, mostly because multiple correct translations are always possible. Ultimately, this can be avoided by conducting a manual evaluation, which is, on the other hand, slow and expensive. Because of that, we only work with automatic metrics in this thesis.

MT evaluation is an intensively studied problem. New metrics are annually evaluated in the WMT share tasks (Neves et al., 2018). In this thesis, we only work with the two most commonly used evaluation metrics: BLEU and METEOR.

BLEU (Bilingual Evaluation Understudy; Papineni et al., 2002) is the most commonly used MT evaluation metric, presumably because of its simplicity. The BLEU score is based on computing frequencies of overlapping word n-grams in system output and references sentences.

The n-gram precision values p_n are computed (for unigrams, bigrams, trigrams and, 4-grams) as a proportion of n-grams in the system output which are present in the references sentences.

To avoid maximizing the precision by generating short outputs, the metric introduces a corpus-level statistic called the *brevity penalty*:

$$\mathrm{BP} = \begin{cases} 1 & h > c \\ \exp\left(1 - \frac{r}{c}\right) & \text{otherwise} \end{cases} \tag{4.1}$$

where r is a number of words in the corpus of reference sentences and c is a number of words in the system output.

The final score is

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\frac{1}{4}\sum_{n=1}^{4} p_n\right). \tag{4.2}$$

Note that the n-gram precisions are averaged on the corpus level and the brevity penalty is a corpus-level statistic as well. BLEU is thus a corpus-level metric and cannot be factorized over sentences.

Originally, the BLEU score was designed to work with multiple reference sentences for a single source sentence. Under these circumstances, it manifests high correlation with human judgment. However, in practice, only one reference sentence is usually available, which is also the case of this thesis.

METEOR (Metric for Evaluation of Translation with Explicit ORdering; Banerjee and Lavie, 2005; Denkowski and Lavie, 2011) is an evaluation metric based on measuring word precision and recall.

The key component of the METEOR score is monolingual word-level alignment between the system output and the reference sentence. The main idea behind the metric is: the more words from the system output are aligned with reference, the better the translation quality.

The alignment algorithm aligns first the exactly same n-grams in the output and the reference, then it aligns n-grams which are the same after stemming, and finally, it uses a language-specific n-gram paraphrase table. The alignment is computed in such a way that every word is matched at most once. The number of covered words is maximized, the number of contiguous chunks is minimized, and the difference of absolute positions in system output and reference is minimized. The use of the paraphrases during the alignment should tackle the problem of not having multiple references.

The final score is computed as a product of a disfluency score d and an adequacy score a. The disfluency score is computed as

$$d = \frac{1}{2}\left(\frac{\text{\# alignment steps}}{\text{\# unigrams matched}}\right)^3 \tag{4.3}$$

and captures how much the system output sentence would need to be split apart in order to be aligned with the reference.

In more recent versions (Denkowski and Lavie, 2011; **?**), METEOR distinguishes between function words and content words. Content words are included in the precision and recall with a higher weight than the function words. The weights are tuned for every target language.

Lexical adequacy is computed as a weighted harmonic mean of precision and recall:

$$a = \frac{10 \cdot P \cdot R}{R + 9P} \tag{4.4}$$

where P and R stand for precision and recall of the output words computed over the monolingual alignment. The final score is the arithmetic average over the evaluated sentences.

For clarity of the presented results, we follow the conventions in the field and report BLEU and METEOR as percentages.

4.1.3 Bringing in Multimodality

MMT is defined as automatic translation of image captions from one natural language into another when using both the source sentence and the image as the system input.

Image captions are a specific text genre. In a coherent text, the textual context of relatively long sentences or even whole documents can contribute to resolving potential ambiguities. This is not true in the case of image descriptions. They are usually short, and thus more prone to suffer from ambiguity. On the other hand, most of the content words in the captions refer to something in the image. If this denotation relation is either implicitly or explicitly captured in the MT model, it can help to achieve better translation of image captions. The learned representations can be used in image retrieval and for theoretical study of how denotation can be computationally grasped.

We can think of many examples where the visual context can play a role in disambiguation, both grammatical and factual. When translating from English to a language that has gendered nouns, the image can be used to decide whether to use a masculine or a feminine form. Without visual grounding, the best option the system choose is to prefer the variant that was more frequent in similar context in the

En: A baseball player in a black shirt just tagged a player in a white shirt.
Fr: Une joueuse de baseball en maillot noir vient de toucher une joueuse en maillot blanc.

The source English sentence misses the gender information necessary for the French translation.

En: A woman sitting on a very large rock smiling at the camera with trees in the background.
De: Eine Frau sitzt vor Bäumen im Hintergrund auf einem sehr großen Stein und lächelt in die Kamera.

The word *rock* is ambiguous. It can be translated as *Stein* (a stone) or *Felsen* (a crag).

En: A boy in a red suit plays in the water.
Cs: Chlapec v červených plavkách si hraje ve vodě.

The word *suit* can be translated into Czech as *oblek* (lounge suit) or *plavky* (swimming suit).

Figure 4.1: Examples of sentences from Multi30k dataset which might be ambiguous without providing the visual evidence.

training data. The second group of examples where visual grounding might help are content words naming object which are not distinguished in the source language but are in the target language. This is shown in Figure 4.1 on examples from the Multi30k dataset.

The examples suggest that if we design a model architecture that is able to utilize both textual and visual input, the model optimization should find a way to represent them. Although the examples seem to point out serious issues for the translation quality, a user study (Frank et al., 2018) showed that resolving the ambiguities only has a limited impact on how users perceive the translation quality and has a minor influence on automatic MT metrics. Proper evaluation of MMT is also a challenging problem.

The user study (Frank et al., 2018) also showed that users are not able to distinguish whether an image caption was written already in the presented language or was translated from a different language. This observation also creates important practical motivation. A potential use case of multimodal translation is translation of manuals with instructional images. Additionally, MMT can find use in digital humanities when making an available image collection captioned in a less frequently spoken language (Elliott and Kleppe, 2016).

4.2 Multi30k Dataset

The machine-learning-based approaches to solve the MMT task require a specialized dataset consisting of images, image captions in a source language and their translations to a target language. In the following sections, we describe the Multi30k dataset (Elliott et al., 2016) first used for the WMT16 shared task. Later, we describe the Czech version of the dataset that we have prepared for the WMT18 shared task.

4.2.1 German and French Versions of Multi30k

Multi30k is a standard dataset used for MMT (Elliott et al., 2016). It is an extension of the Flickr30k dataset (Young et al., 2014) for English image captioning.

The original Flickr30k dataset consists of photos capturing mostly people posing for pictures and during various activities, but also sports events, pets or outdoor urban and landscape scenes. The images were collected from albums that Flickr users made publicly available under the Creative Commons license. All images are accompanied by five independent English captions that should describe what is in the image, usually in general terms, trying to avoid using specific knowledge of places and people displayed in the pictures.

The training set consists of 29,000 images, the validation set contains 1,000 images, and 1,041 images are left for testing. The Multi30k dataset uses the same splits as Flickr30k. One caption for each image is translated into German, French, and Czech. The German captions were produced by professional translators, the French and Czech versions were crowd-sourced.

For the 2017 and the 2018 competitions, new test sets of 1,000 images were created. The preparation of the test sets followed the same methodology as was used for collecting the original Flickr30k dataset. Additionally, a smaller dataset focused on ambiguous words was created from the MS COCO dataset. The images were

split	sentences	English			German		
		tok./sent.	ch./tok.	OOV	tok./sent.	ch./tok.	OOV
train	29,000	13.0	4.8	—	12.4	5.9	—
validation	1,014	13.1	4.8	1.2%	12.7	6.0	3.0%
test 2016	1,000	13.0	4.9	1.1%	12.1	5.9	2.6%
test 2017	1,000	11.3	4.8	1.8%	10.8	6.1	4.1%
test COCO	461	11.3	4.9	1.1%	11.2	5.9	3.7%
		French			Czech		
		tok./sent.	ch./tok.	OOV	tok./sent.	ch./tok.	OOV
train	29,000	14.1	5.4	—	10.2	6.0	—
validation	1,014	14.2	5.4	1.2%	10.2	5.9	3.9%
test 2016	1,000	14.0	5.5	1.1%	10.5	6.0	4.0%
test 2017	1,000	12.6	5.5	1.6%			
test COCO	461	12.4	5.6	1.6%			

Table 4.1: Statistics of the Multi30k dataset for different languages that include the average number of tokens per sentence (tok./sent.), the average number of characters per token (ch./tok.) and out-of-vocabulary rate (OOV) with respect to the training set.

taken from a subset of the MS COCO dataset that is included in the VerSe dataset (Gella et al., 2016) where the captions were annotated with the OntoNote word senses (Pradhan et al., 2007). Using these annotations, 416 images with captions containing potentially ambiguous words were selected.

Statistics of the dataset are tabulated in Table 4.1. The statistics correspond to typological differences between the used languages. Whereas English and French have sentences of similar length (in terms of a number of tokens), German and Czech sentences are shorter. On the other hand, both the languages have a much higher out-of-vocabulary rate, in German because of compounding, in Czech due to the rich inflection. Note also that even though the 2017 test set was collected using a similar protocol as the rest of the data, it has on average shorter sentences and higher out-of-vocabulary rate which suggests that the methodology of collecting the data differs in some important details.

Because of that and to make results of our experiments in Chapter 5 comparable even with the early work on MMT, we report all our results on the 2016 test set.

In addition to the translation data, the images are also accompanied by 5 independently created crowd-sourced German captions. These were originally intended for experiments with cross-lingual image captioning, a task of generating (German) image captions. At test time, only the image is available as a system input, whereas during training, captions in multiple languages (both English and German) can be used.

The dataset has several shortcomings. A relatively small number of sentences, limited vocabulary and similar structure of most sentences make it easy to achieve a good translation quality with phrase-based models or neural models with only textual input (Specia et al., 2016). The statistics of linguistic phenomena in the dataset and the comparison with the general domain data is tabulated in Table 4.2. There were also cultural and ethical issues identified in the underlying Flickr30k dataset (Miltenburg, 2016) which is burdened with racial and gender stereotypes.

4.2.2 Czech Version of Multi30k

The original WMT shared task in MMT in 2016 was organized with English-to-German translation only. In 2017, French was added as another target language. For the 2018 competition, we also created a Czech version of the dataset. Our motivation was that Czech as a morphologically rich language might be a more challenging target language given the limited vocabulary.

The translation was conducted using the same methodology and software as the French translation in 2017. The translators were presented with the images in a random order, and they always saw the image and the English caption at the same time and were asked to provide the Czech translation.

Most of the translation was conducted by high school and university students, native speakers of Czech with a good command of English. The test data was translated by English teachers and is thus supposed to have higher quality than the rest of the dataset. This resulted in a slight domain mismatch because the high school students preferred to use informal language whereas the translations conducted by the teachers tend to be more formal.

The dataset was automatically checked for:

- Mismatching punctuation at the end of the sentence;
- Spelling errors using GNU Aspell[1];
- Missing punctuation between clauses;

[1] `http://aspell.net/`

		Multi30k	CzEng
corpus statistics	sentence count	29,000	1,163,584
	unique tokens	9,795	196,519
	unique tokens at least 5 times	2,903	43,563
	unique tokens at least 10 times	1,967	29,732
	unique lemmas	7,333	183,011
	unique lemmas at least 5 times	2,387	36,458
	unique lemmas at least 10 times	1,651	24,311
	sentence lengts	13.1 ± 4.1	13.2 ± 15.8
morphology and syntax	number of clauses	1.5 ± .7	1.3 ± .8
	auxiliary verb ratio	2.6%	2.9%
	content verb ratio	11.5%	14.4%
	noun ratio	30.2%	14.7%
	pronoun ratio	0.4%	8.2%
	adjective ratio	9.0%	6.5%
	adverb ratio	1.1%	5.9%
	numeral ratio	1.9%	1.5%
	subject 'I' frequency	0.1%	15.6%
	subject 'you' frequency	0.1%	13.0%
	subject 'he'/'she'/'it' frequency	1.3%	13.2%
	subject 'we' frequency	0.1%	4.2%
	subject 'they' frequency	0.6%	2.7%
	subject 'there' frequency	0.8%	1.8%
	subject 'this' frequency	0.2%	1.6%
	contains modal verb	0.2%	17.6%
	contains past tense	13.3%	36.2%
	contains future tense	0.0%	1.9%
	contains conditional	0.0%	1.8%
	is question	0.0%	13.0%
named entities	entities per sentence	.323	.672
	person	.009	.115
	nationalities and other groups	.028	.021
	facilities (buildings, airports, highways, ...)	.001	.003
	organization (companies, agencies, institutions)	.018	.189
	countries, cities, states	.009	.060
	other locations	.002	.008
	product	.003	.008
	event (hurricanes, battles, sports events)	.000	.003
	work of art	.001	.004
	law (named documents)	.000	.021
	language	.000	.001
	date	.013	.079
	time	.006	.016
	percent	.000	.008
	money	.000	.003
	quantity	.002	.005
	ordinal number	.002	.018
	cardinal number	.228	.111

Table 4.2: Statistics of linguistic features of the English side of the Multi30k dataset compared with a 1 million sample of the CzEng parallel corpus computed using spaCy (`https://spacy.io`).

	Proportion of data	Annotator agreement
No spelling errors	94%	92%
Stylistically appropriate	75%	73%
Adequate in meaning	96%	93%
No inappropriate lexical anglicism	94%	90%
No inappropriate syntactic anglicism	93%	91%

Table 4.3: Error analysis on the Czech version of the dataset.

- Suspiciously short and long sentences when compared to the source sentence length;
- Characters which are neither in the Czech alphabet and neither are punctuation.

The sentences where any of the errors were spotted were manually corrected. In total, 5,255 sentences (16%) were manually corrected, mostly for spelling and grammar errors.

During the manual checking, we encountered many cases with lexical or syntactic anglicism which did not sound as fluent Czech sentences. After the manual correction, we randomly sampled 1% of the sentences and manually annotated the quality of the translations. Three people annotated the data, every sentence was annotated by two different people. The results are tabulated in Table 4.3. The sentences are mostly (over 90% sentences, with over 90% annotators agreement) without spelling errors and adequate in meaning. One quarter of the sentences was marked as stylistically inappropriate, however, the annotator agreement is only 73% for this category.

4.3 Model Architectures for Multimodal Translation

In this section, we provide an overview of existing models for MMT and classify the approaches into groups based on what form of visual information they exploit and how the visual information is plugged into the models. The section briefly mentions some of our models which are discussed in more detail in Sections 5.1 and 5.2. For a fair comparison, we report only results achieved in the constrained setup, i.e., while using only Multi30k dataset for training.

The approaches to MMT can be categorized based on the way the image features are incorporated into the translation system. Even though there are attempts to exploit the visual information within the phrase-based paradigm (Koehn et al., 2007), most of the MMT systems follow the encoder-decoder scheme from neural MT. There are four main ways of using visual information:

- Use the image to re-rank an n-best list from a text-only system (either neural or phrase-based);
- Use the visual input on the source side (as a part of the encoder) and pass multimodal representation from the encoder to the decoder;
- Use the visual input on the target side and combine independent visual and textual inputs in the decoder;
- Use the visual grounding only as an auxiliary objective during the model training.

Another criterion we can use to categorize the models is in what form the image features are incorporated. These can be:

- Explicit object recognition;
- A fixed-sized vector representation summarizing the image (the penultimate layer of an image classification network);
- The model can attend different image areas (convolutional map) with attention mechanism independently.

Shah et al. (2016) used a hybrid approach based on a statistical MT system (Koehn et al., 2007) trained on the Multi30k data only. The n-best list produced by an MT system is re-ranked using a probability distribution over objects from an image classification network together with features from the MT system.

The first purely neural approach we know of (Elliott et al., 2015) used the Recurrent Neural Network (RNN) encoder-decoder setup without attention (Sutskever et al., 2014). The image vector is used to initialize both encoder and decoder. The multimodal setup clearly surpasses the text-only model, but the translation quality is by large margin worse than results achieved later with the attention model (Bahdanau et al., 2014) or even the phrase-based systems (Koehn et al., 2007).

Calixto and Liu (2017) experimented with plugging the last fully connected layer of the VGG-19 network (Simonyan and Zisserman, 2014) in the attentive RNN sequence-to-sequence model (Bahdanau et al., 2014). The decoder only attends to the hidden states of the text encoder. The authors explored encoder and decoder initialization with a projection of the image representation. The best results are achieved

with the decoder initialization. The paper reports significant improvement over the text-only baseline. However, the presented results are lower than the results of similar experiments conducted by Caglayan et al. (2017) where the multimodal combination did not improve the translation quality of the text-only systems.

The approach by Calixto and Liu (2017) is similar to our first system for the WMT competition (Libovický et al., 2016). Besides initializing the decoder with the image vector, we also used output of a phrase-based MT (Koehn et al., 2007) system with an additional Language Model (LM) based on coarse bi-token classes (Stewart et al., 2014) as additional input. MMT is in this case treated as an MT post-editing task.

Caglayan et al. (2017) in the winning submission to the WMT17 shared task, systematically explored various strategies how the convolutional map from ResNet (He et al., 2016) can be projected into a single vector and plugged into an RNN sequence-to-sequence model (Bahdanau et al., 2014). The strategies include encoder and decoder initialization, and point-wise gating of the attention context vector and target embeddings. Their results do not show any significant differences from the text-only model. A similar approach was taken also by Madhyastha et al. (2017) who tried to use a posterior probability distribution instead of the image representation from intermediate layers.

A combination of explicit object detection and image vector representation was used by Huang et al. (2016), the WMT16 shared task winners. Their system first detects the three largest objects in the image. Then, the representation from the penultimate layer of the VGG-19 network for both the original image and the cropped areas are used to initialize four independent sentence encoders. All the encoders are attended in the decoder that concatenates the context vectors before computing the next token probabilities.

Caglayan et al. (2016) presented a doubly attentive model performing the attention independently over the image and the source sentence. The authors make a strong assumption that the network can be trained in such a way that the hidden states of the encoder and the projected states of the convolutional network occupy the same vector space and thus sum the context vectors from both modalities. In this way, their MMT system remains far below the text-only setup.

Similarly, Calixto et al. (2017) used two independent attention mechanisms and concatenated the context vectors. The authors claimed that the multimodal model significantly surpasses the text-only model. However, their baseline model achieves lower scores than Caglayan et al. (2016) while using almost the same architecture.

An approach that uses the visual information at the training time only was introduced by Toyama et al. (2016). Their system uses a variational autoencoder (Kingma et al., 2014) approach to generate a latent variable that should capture the visual grounding of the source sentence. At the inference time, the image is no longer necessary because the latent variable is inferred using the encoder states only.

The Imagination model (Elliott and Kádár, 2017) exploits the same idea but applies it in a fully discriminative multi-task learning setup. This model is an RNN-based encoder-decoder architecture for MT with an auxiliary output which is an image representation from the penultimate layer of an image classification network. Following Chrupała et al. (2015), the model generates also an estimated image representation computed using regression from the mean-pooled encoder states. The main advantage of this approach is that it can make use of additional training data for both translation and image representation. Although such a model is not capable of directly using the input image for word disambiguation, it was able to significantly improve the translation quality by using an order of magnitude larger data, namely the NewsCommentary parallel corpus (Tiedemann, 2012) and English image captions from the MS COCO dataset (Chen et al., 2015).

Current state-of-the-art results were achieved by Grönroos et al. (2018). Their multimodal system is based on a pre-trained Transformer model trained on a large parallel corpus consisting of sentence-pairs filtered from the OpenSubtitles corpus (Tiedemann, 2012) and synthetic target sentences produced by a text-only system trained on large data. The encoder is then given a special token indicating whether the target sentence is an automatic or a manual translation. With this domain adaptation technique, the model already outperforms all other MMT models. To add the multimodal information in the model, the authors introduced a gating mechanism that modifies the logits over the vocabulary using the image representation and current state of the decoder. The last step, however, only provides a negligible improvement.

A summary of the approaches categorized according to the discussed criteria is provided in Table 4.4. In the case of the non-attentive RNN encoder-decoder model without attention, the visual information brings substantial improvement (Elliott et al., 2015). With stronger attentive models, the contribution of multimodal information is often not significant. Even though some papers report improvements using multimodal models (Calixto and Liu, 2017; Calixto et al., 2017), similar approaches with stronger text-only baselines do not confirm these results (Caglayan et al., 2016, 2017; Libovický et al., 2016; Helcl and Libovický, 2017a).

method	image place in the model	image information type	BLEU	
Moses (Specia et al., 2016)	—	—	34.6	×
RNN, text only	—	—	36.7	∗
Transformer, text only	—	—	38.3	∗
Elliott et al. (2015)	enc., dec.	vector	23.1	
Shah et al. (2016)	reranking	vector	34.8	×
Huang et al. (2016)	encoder	vector	36.8	×
Caglayan et al. (2016)	decoder	conv.	36.2	×
Caglayan et al. (2017)	decoder	conv.	37.8	
Calixto and Liu (2017)	decoder	vector	37.3	
Calixto et al. (2017)	decoder	conv.	36.5	
Libovický and Helcl (2017), Section 5.1	decoder	conv.	37.6	∗
Toyama et al. (2016)	auxiliary	vector	36.5	
Elliott and Kádár (2017)	auxiliary	vector	36.8	
Libovický et al. (2018a), Section 5.2	decoder	conv.	38.5	
Helcl et al. (2018b), Section 5.3	auxiliary	vector	39.2	
the same model in unconstrained setup	auxiliary	vector	42.6	
Grönroos et al. (2018), unconstrained	decoder	vector	45.1	

Table 4.4: An overview of the methods used for MMT. The BLEU scores are reported on English-to-German translation, for constrained (trained on Multi30k only), single-model setup on the 2016 test set. Results marked with '∗' were computed for this thesis. Results marked with '×' were obtained from the WMT16 submissions provided by the WMT organizers and re-evaluated with tokenization that was used in the later WMT competitions. The values thus differ from what was reported in teh original papers. The other results are reported as they were originally published. Grönroos et al. (2018) did not report a model trained on Multi30k only, therefore we compare it with an unconstrained version of our best system (both in gray color).

In 2017 (Elliott et al., 2017) and 2018 (Barrault et al., 2018), the WMT shared task included human evaluation of the submitted systems. In most of the cases, the results of the human evaluation agree with the automatic metrics. In the 2017 evaluation, the results were in favor of the multimodal systems which may suggest that the multimodal systems might help the translation in aspects which are not captured with the automatic evaluation metrics.

5

Architectures for Multi-Source Sequence-to-Sequence Learning

In this chapter, we present our original contributions to solving Multimodal Machine Translation (MMT) which lie mostly in the model architecture design.

In Section 5.1, we present a modification of the Recurrent Neural Network (RNN) sequence-to-sequence models that allows the decoder to attend multiple sources at once in an interpretable way. Section 5.2 introduces similar modifications made in the Transformer architecture. In both cases, we conduct experiments on MMT and one additional task to prove general usability of the proposed architectures.

Additionally, Section 5.3 summarizes experiments with additional data and multi-task learning that further improve the quality of the MMT model outputs.

5.1 Modality Combination in Recurrent Sequence-to-Sequence Models

*This section is based on the paper "*Attention Strategies for Multi-Source Sequence-to-Sequence Learning*", joint work with Jindřich Helcl published at ACL 2017.*

In this section, we introduce a modification to the attentive sequence-to-sequence models (Bahdanau et al., 2014) that allows working with multiple inputs while generating a single output sequence. In attentive RNN sequence-to-sequence models with multiple inputs, the decoder needs to combine information collected from the encoders. We describe a novel architecture that we have developed to tackle this

problem. Unlike previous work, our techniques explicitly model distribution over the inputs and is more interpretable. We evaluate the method not only on the MMT but also on the Automatic MT Post-Editing (APE) task to show it can be used in general multi-source setup.

In the single-source setup, the attention is formulated in the following way. We denote $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_{T_x}) \in \mathbb{R}^{T_x \times d_h}$ the input states of the encoder of dimension d_h, T_x input length, T_y ground-truth output length and $\mathbf{s}_i \in \mathbb{R}^{d_s}$ states of the decoder of dimension d_s. As already described in Section 2.3.3, in the single-input setup, the attention mechanism with intermediate state size d_a works as follows:

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_i + \mathbf{U}_a \mathbf{h}_j + \mathbf{b}_a) + b_e, \tag{5.1}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{5.2}$$

$$\mathbf{c}_i = \sum_{j=1}^{T_x} \alpha_{ij} \mathbf{h}_j. \tag{5.3}$$

The *energies* $e_i \in \mathbb{R}^{T_x}$ are computed as a projection of the input states and the current decoder state. They are normalized using a softmax to the *attention distribution* α_i. The distribution is used to compute a weighted sum over the input states which is called the *context vector* $\mathbf{c}_i \in \mathbb{R}^{d_h}$.

In the following description, different inputs to the decoder are denoted by upper indices in brackets. Assume that the decoder can use N independent inputs $\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)}$. Unlike Chapter 2, the upper index is used to distinguish the model inputs, not layers in the network. The simplest solution for combining two attention models in the decoder is a concatenation of the context vectors $\mathbf{c}_i^{(1)}, \dots, \mathbf{c}_i^{(N)}$ of N inputs (Zoph and Knight, 2016; Firat et al., 2016). In this setting, the decoder attends to all inputs independently and the interaction of the inputs is resolved implicitly in the decoder layers that follow the context vector.

Instead, we propose two alternative strategies for attention combination when using multiple inputs and a single decoder. We either let the decoder learn the α_i distribution jointly over all inputs states (*flat* attention combination) or we factorize the distribution over individual encoders (*hierarchical* combination). Both of the methods explicitly compute a distribution over the inputs. The distribution tells how much attention is paid to each input at every step of the decoder.

Additionally, we experiment with the *sentinel gate* (Lu et al., 2017), an extension of the attentive RNN decoder with Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) originally introduced in the context of image captioning. We adjust the mechanism for Gated Recurrent Unit (GRU) networks (Cho et al., 2014a), which we use in our experiments. The gate applied over state $\mathbf{s}_i \in \mathbb{R}^{d_s}$ is computed as:

$$\psi_\mathbf{i} = \sigma(\mathbf{W}_y \mathbf{y}_i + \mathbf{W}_s \mathbf{s}_{i-1}) \tag{5.4}$$

where $\mathbf{W}_y \in \mathbb{R}^{d_e \times s}$ and $\mathbf{W}_s \in \mathbb{R}^{d_s \times d_s}$ are trainable parameters, $\mathbf{y}_i \in \mathbb{R}^{d_e}$ is the embedded decoder input of dimension d_e, and $\mathbf{s}_{i-1}$ is the previous decoder state.

Analogically to Equation 5.1, we compute a scalar energy term for the sentinel gate:

$$e_{\psi_i} = \mathbf{v}_a^\top \tanh\left(\mathbf{W}_a^{(\psi)} \mathbf{s}_i + \mathbf{U}_a^{(\psi)} (\psi_{i-1} \odot \mathbf{s}_i) + \mathbf{b}_a^{(\psi)}\right) + b_e^{(\psi)} \tag{5.5}$$

where $\mathbf{W}_a^{(\psi)} \in \mathbb{R}^{d_s \times d_a}$, $\mathbf{U}_a^{(\psi)} \in \mathbb{R}^{d_s \times d_a}$ are the projection matrices, $\mathbf{v}_a \in \mathbb{R}^{d_a}$ is the weight vector, $\mathbf{b}_a^{(\psi)} \in \mathbb{R}^{d_a}$ and $b_e \in \mathbb{R}$ biases, and $\psi_i \odot s_i$ is the sentinel vector. Note that the sentinel energy term does not depend on any of the inputs. The sentinel vector is projected to the same vector space as the encoder state $\mathbf{h}_j$ in Equation 5.1. The term e_{ψ_i} is added as an extra attention energy term to Equation 5.2 and the sentinel vector $\psi_\mathbf{i} \odot \mathbf{s}_i$ is used as the corresponding vector in the summation in Equation 5.3.

This technique allows the decoder to choose whether to attend to the input or utilize the information that is already in the decoder state and thus act more like a Language Model (LM).

5.1.1 Proposed Strategies

Flat. In the flat attention combination, we first project the states of all inputs into a shared space and then compute a distribution over the projections. The main difference between the context vector concatenation and the flat attention combination is that the $\alpha_i \in \mathbb{R}^{\sum_k T_x^{(k)}}$ coefficients are computed jointly for all inputs:

$$e_{ij}^{(k)} = \mathbf{v}_a^\top \tanh\left(\mathbf{W}_a \mathbf{s}_i + \mathbf{U}_a^{(k)} \mathbf{h}_j^{(k)} + \mathbf{b}_a^{(k)}\right) + b_e^{(k)} \tag{5.6}$$

$$\alpha_{ij}^{(k)} = \frac{\exp(e_{ij}^{(k)})}{\sum_{n=1}^{N} \sum_{m=1}^{T_x^{(n)}} \exp\left(e_{im}^{(n)}\right)} \tag{5.7}$$

where $T_x^{(k)}$ is the length of the k-th input sequence and $e_{ij}^{(k)} \in \mathbb{R}$ is the attention energy of the j-th state of the k-th encoder in the i-th decoding step.

The attention energies are computed in the same way as in Equation 5.1. The parameters $\mathbf{v}_a \in \mathbb{R}^{d_a}$ and $\mathbf{W}_a \in \mathbb{R}^{d_s \times d_a}$ are shared among the inputs, and $U_a^{(k)} \in \mathbb{R}^{d_h^{(k)} \times d_a}$ is trained independently for each input and serves as an input-specific projection of hidden states into a common vector space.

The states of the individual inputs may have a different dimensionality and the dimensions may have a different meaning, especially when we use pre-trained networks for different modality representation. The context vector cannot be computed as their weighted sum. Instead, we first project all the input states into a common space using linear projections:

$$c_i = \sum_{k=1}^{N} \sum_{j=1}^{T_x^{(k)}} \alpha_{ij}^{(k)} \mathbf{U}_c^{(k)} \mathbf{h}_j^{(k)} + \mathbf{b}_c^{(k)} \tag{5.8}$$

where $\mathbf{U}_c^{(k)} \in \mathbb{R}^{d_h^{(k)} \times d_a}$ and $\mathbf{b}_c^{(k)} \in \mathbb{R}^{d_a}$ are additional trainable parameters.

The matrices $\mathbf{U}_c^{(k)}$ should project the hidden states into a common space. From the architecture design, it is not clear whether these projections can be the same as the one used in the energy computation using matrices $\mathbf{U}_a^{(k)}$ in Equation 5.1, i.e., whether $\mathbf{U}_c^{(k)} = \mathbf{U}_a^{(k)}$. In our experiments, we explore both options. We also try both adding and not adding the sentinel $\alpha_i^{(\psi)} \mathbf{U}_c^{(\psi)} (\psi_i \odot \mathbf{s}_i)$ to the context vector.

Hierarchical. In the hierarchical attention combination, we first compute a context vector for each input independently, similarly to the concatenation approach. Instead of concatenating them, we perform a second attention mechanism over the context vectors where we treat the context vectors as states of an encoder.

The hierarchical attention is thus computed in two steps. In the first step, we compute the context vector $\mathbf{c}_i^{(k)}$ for each encoder independently using Equation 5.3. In the second step, we project the context vectors (and optionally the sentinel) into a common space and compute another distribution β_i over the projected context vectors and their corresponding weighted average $\mathbf{c}_i$:

$$e_i^{(k)} = \mathbf{v}_b^\top \tanh\left(\mathbf{W}_b \mathbf{s}_i + \mathbf{U}_b^{(k)} \mathbf{c}_i^{(k)} + \mathbf{b}_b^{(k)}\right) + b_f^{(k)} \tag{5.9}$$

$$\beta_i^{(k)} = \frac{\exp(e_i^{(k)})}{\sum_{n=1}^{N} \exp(e_i^{(n)})}, \tag{5.10}$$

$$\mathbf{c}_i = \sum_{k=1}^{N} \beta_i^{(k)} \mathbf{U}_c^{(k)} \mathbf{c}_i^{(k)} + \mathbf{b}_c^{(k)} \tag{5.11}$$

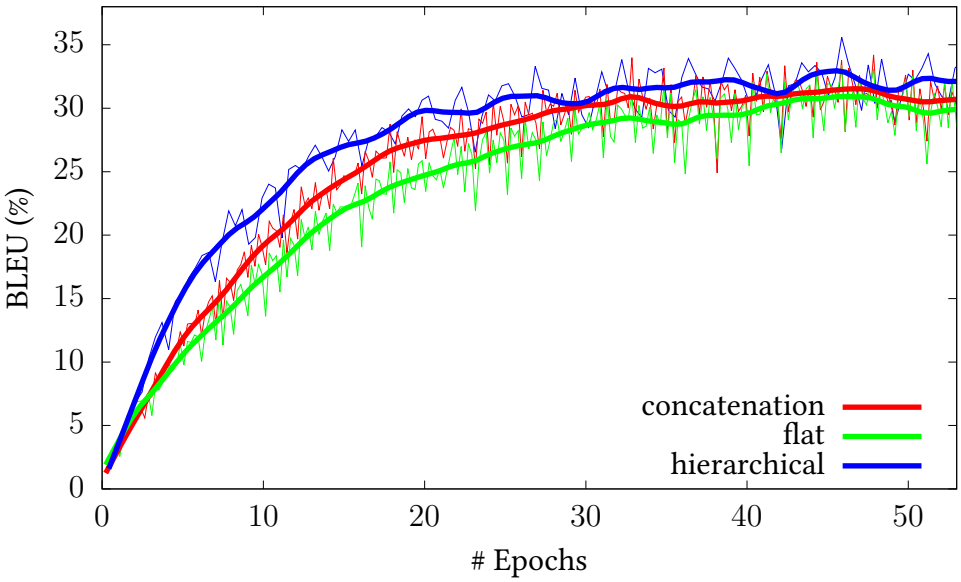


Figure 5.1: Learning curves for German MMT with RNN-based models on validation data for context vector concatenation (blue), flat (green) and hierarchical (red) attention combination without sentinel and without sharing the projection matrices. To make the trends more clearly visible, the learning curves are smoothed using the cubic spline.

where $\mathbf{c}_i^{(k)} \in \mathbb{R}^{d_h^{(k)}}$ is the context vector computed from the k-th input. The additional trainable parameters $\mathbf{v}_b \in \mathbb{R}^{d_a}$ and $\mathbf{W}_b \in \mathbb{R}^{d_s \times d_a}$ are shared for all encoders, matrices $\mathbf{U}_b^{(k)} \in \mathbb{R}^{d_h^{(k)} \times d_a}$ and $\mathbf{U}_c^{(k)} \in \mathbb{R}^{d_h^{(k)} \times d_a}$ are used for encoder-specific projection that can be shared, similarly to the case of flat attention combination.

5.1.2 Experiments with Multimodal Translation

The models were implemented using Neural Monkey, a framework for sequence-to-sequence learning (Helcl and Libovický, 2017b; Helcl et al., 2018a).[1]

We process the textual input with bidirectional GRU network (Cho et al., 2014a) with 300 units in the hidden state in each direction and 300 units in embeddings. For the attention projection space, we use 500 hidden units. We optimize the network to minimize the output cross entropy using the Adam algorithm (Kingma and Ba, 2014) with learning rate of 10^{-4}.

In the original experiments for ACL 2017, the visual input is processed with a pre-trained VGG-16 network (Simonyan and Zisserman, 2014) without further fine-tuning. Attention distribution over the visual input is computed from the last convolutional layer of the network. The decoder utilizes conditional GRU units (Firat

[1] `http://github.com/ufal/neuralmonkey`

and Cho, 2016) with both layers of dimension 500. We use Byte-Pair Encoding (BPE) (Sennrich et al., 2016b) with a vocabulary of 20,000 sub-word units shared between the text encoder and the decoder. The target sentences are decoded greedily. The experiments were conducted only with the English-to-German translation.

For a better comparison with more recently published results, we also present another set of experiments. We replaced the VGG-16 network with ResNet (He et al., 2016) which showed superior performance in many tasks combining language and vision (Elliott and Kádár, 2017; Zhu et al., 2018). During decoding, we use beam search of width 10 and length normalization of 1.0 (Wu et al., 2016). These experiments cover all 3 target languages present in the Multi30k dataset. We do not repeat experiments with sentinel gate and sharing the projection matrices which led to substantially worse performance.

During the evaluation, we follow the preprocessing used in Workshop of Machine Translation (WMT) Multimodal Translation Shared Task (Specia et al., 2016). The sentences are tokenized with the Moses tokenizer with normalized punctuation and lowercased. We evaluate the results using BLEU (Papineni et al., 2002) and METEOR (**?**) as implemented in MultEval[2] which estimates the confidence intervals for the scores using bootstrap resampling (Koehn, 2004).

Results of related work show (Elliott and Kádár, 2017) that the increased translation quality of the multimodal models compared to text-only models can come from enhancing the input representation, or the image may only introduce noise and act as a regularizer. In order to test whether it is also the case with our models or whether our models explicitly use the visual input, we perform an adversarial evaluation similar to Elliott (2018). We evaluate the model when providing a random image as an input unrelated to the source sentence and observe how it affects the scores and whether providing the incorrect image influences the translation quality.

The results of the original experiments are shown in Table 5.1. None of the multimodal models outperforms the text-only models. The best multimodal results are achieved using the hierarchical attention combination without the sentinel mechanism, which also shows the fastest convergence (see learning curves in Figure 5.1). The flat combination strategy achieves similar results eventually. Sharing the projections for energy and context vector computation does not improve over the concatenation baseline and slows down the training convergence almost prohibitively.

Compared to a single-layer GRU, using the conditional GRUs brought an improvement of about 1.5 BLEU points on average, except for the concatenation scenario where the translation quality dropped by almost 5 BLEU points. We hypothesize this is caused by the fact the model has to learn the implicit attention combination

[2] `https://github.com/jhclark/multeval`

	shared proj.	sentinel	MMT		APE	
			BLEU	METEOR	BLEU	HTER
baseline			32.4	49.3	62.3	24.8
concatenation			31.4 ± .8	48.0 ± .7	62.3 ± .5	24.4 ± .4
flat	×	×	30.2 ± .8	46.5 ± .7	62.6 ± .5	24.2 ± .4
	×	✓	29.3 ± .8	45.4 ± .7	62.3 ± .5	24.3 ± .4
	✓	×	30.9 ± .8	47.1 ± .7	62.4 ± .6	24.4 ± .4
	✓	✓	29.4 ± .8	46.9 ± .7	62.5 ± .6	24.2 ± .4
hierarchical	×	×	**32.1 ± .8**	**49.1 ± .7**	62.3 ± .5	24.1 ± .4
	×	✓	28.1 ± .8	45.5 ± .7	62.6 ± .6	24.1 ± .4
	✓	×	26.1 ± .7	42.4 ± .7	62.4 ± .5	24.3 ± .4
	✓	✓	22.0 ± .7	38.5 ± .6	62.5 ± .5	24.1 ± .4

Table 5.1: Results of our experiments on the test sets of Multi30k dataset and the APE dataset as originally published (Libovický and Helcl, 2017). The column 'shared proj.' denotes whether the projection matrix is shared for energies and context vector computation.

on multiple places—once in the output projection and three times for each projection matrix inside the conditional GRU (Firat and Cho, 2016, Equations 10-12). We thus report the scores of the flat and the hierarchical attention combination techniques trained with conditional GRUs and compare them with the concatenation baseline trained with plain GRUs.

Results of the re-done experiments are tabulated in Table 5.2. Even with a stronger image representation, we were not able to surpass the text-only baseline. The adversarial evaluation with randomly selected input images shows differences of around 0.5 BLEU points which suggests that the models really utilize the visual information during inference. The difference between the translation quality of the text-only and the multimodal models can be probably due to overfitting. The 2 to 4 BLEU points improvement compared to the original results can be mostly attributed to using BPE-based vocabulary and beam search during decoding.

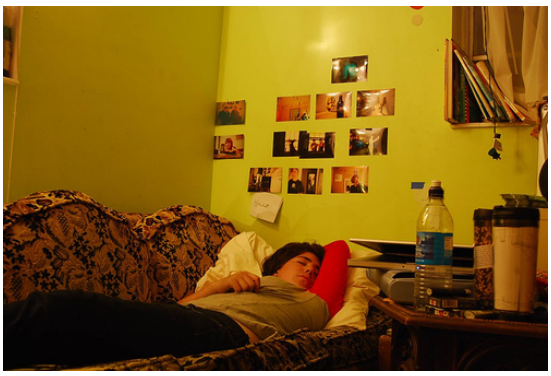
The strongest feature of the attention combination models is their interpretability. Figure 5.2 shows the attention paid to the inputs during the decoding with the hierarchical attention model using the sentinel. We can see that while decoding prepositions and articles, the model pays attention to the sentinel, presumably because the words can be inferred from the language context. Most of the attention is distributed to the textual input which, as the adversarial evaluation suggests. The attention is distributed more towards the image only in cases when it generates names of objects which are present in the image.

English → German	BLEU	METEOR	Adversarial	
			BLEU	METEOR
text-only	36.7 ± .8	55.1 ± .7	—	—
decoder initialization	36.9 ± .8	54.2 ± .6	35.8 ± .8	54.5 ± .6
concatenation	35.7 ± .8	54.4 ± .6	30.9 ± .8	54.7 ± .6
flat	34.6 ± .8	54.3 ± .6	33.8 ± .8	53.7 ± .6
hierarchical	37.6 ± .8	56.0 ± .6	34.2 ± .8	55.2 ± .6

English → French	BLEU	METEOR	Adversarial	
			BLEU	METEOR
text-only	48.3 ± .8	69.5 ± .6	—	—
decoder initialization	48.1 ± .8	68.3 ± .3	48.1 ± .9	69.0 ± .6
concatenation	47.7 ± .8	68.5 ± .6	41.3 ± .8	68.2 ± .6
flat	46.0 ± .9	68.2 ± .6	43.5 ± .8	67.5 ± .6
hierarchical	48.2 ± .9	69.2 ± .6	44.9 ± .8	68,7 ± .6

English → Czech	BLEU	METEOR	Adversarial	
			BLEU	METEOR
text-only	30.0 ± .8	29.1 ± .4	—	—
decoder initialization	29.6 ± .8	29.0 ± .4	29.3 ± .8	28.7 ± .4
concatenation	29.3 ± .8	28.6 ± .4	24.2 ± .8	28.7 ± .4
flat	29.1 ± .8	28.3 ± .4	26.5 ± .8	28.1 ± .4
hierarchical	29.5 ± .8	28.7 ± .4	28.1 ± .8	28.5 ± .4

Table 5.2: Quantitative results of the MMT experiments on the 2016 test set using the RNN models in terms of BLEU and METEOR.



Source: a man sleeping in a green room on a couch .
Reference: ein Mann schläft in einem grünen Raum auf einem Sofa .
Output with attention:

(1) source, (2) image, (3) sentinel

Figure 5.2: Visualization of hierarchical attention in MMT. Each column in the diagram corresponds to the weights of the encoder, the image, and sentinel (Equation 5.10). Note that despite the overall low importance of the image encoder, it gets activated for the content words. The visualization is taken from the original paper (Libovický and Helcl, 2017).

Figure 5.3 shows the attention paid to the visual and the textual input for decoding without the sentinel mechanism. Here again, most of the attention is to the textual input. The image is attended only when the decoder generates a noun. We hypothesize this is because the pre-trained Convolutional Neural Network (CNN) that we used for extracting image features is trained for object recognition where the object often corresponds to nouns in the image descriptions.

5.1.3 Experiments with Automatic Machine Translation Post-Editing

As another use case for our attention combination strategies, we experiment with APE. It is a task of improving automatically generated translation given both the system output and the source sentence. The original translation system is treated as a black box and cannot be modified in any way. The main idea behind this task is

Source: a brown dog is running after the black dog .

Reference de: ein brauner hund rennt dem schwarzen hund hinterher .
Reference fr: un chien brun court après le chien noir .
Reference cs: hnědý pes běží za černým psem .

Output with attention:

Source: a female playing a song on her violin .

Reference de: eine frau spielt ein lied auf ihrem cello .
Reference fr: une femme jouant un morceau sur son violon .
Reference cs: žena hraje píseň na housle .

Output with attention:

Figure 5.3: Visualization of the MMT model with hierarchical attention without the sentinel gate. The diagrams show distribution β_i (Equation 5.10) for (1) source sentence, and (2) the image.

that different Machine Translation (MT) systems based on different principles tend to make different types of errors which might be in some sense complementary. An APE system based on a different principle the primary MT system then can help to fix some of these errors.

Source	Choose Uncached Refresh from the Histogram panel menu.
MT	Wählen$_1$ Sie$_2$ Uncached$_3$ ”$_4$ Aktualisieren$_5$ ”$_6$ aus$_7$ dem$_8$ Menü$_9$ des$_{10}$ Histogrammbedienfeldes$_{11}$.$_{15}$
Reference	Wählen$_1$ Sie$_2$ ”$_4$ Nicht$_{12}$ gespeicherte$_{13}$ aktualisieren$_{13}$ ”$_6$ aus$_7$ dem$_8$ Menü$_9$ des$_{10}$ Histogrammbedienfeldes$_{11}$.$_{15}$
Edit ops.	*keep*$_1$ *keep*$_2$ *delete*$_3$ *keep*$_4$ Nicht$_{12}$ gespeicherte$_{13}$ aktualisieren$_{13}$ *delete*$_5$ *keep*$_6$ *keep*$_7$ *keep*$_8$ *keep*$_9$ *keep*$_{10}$ *keep*$_{11}$ *keep*$_{15}$

Figure 5.4: An example of a sequence of edit operations that our system should learn to produce when given the candidate automatic translation. The colors and subscripts denote the alignment between the edit operations and the machine-translated and post-edited sentence. The example is taken from Libovický et al. (2016).

We used data from the WMT16 APE Task (Bojar et al., 2016a; Turchi et al., 2016), which consists of 12,000 training, 2,000 validation, and 1,000 test sentence triplets from the IT domain. Each triplet contains an English source sentence, an automatically generated German translation of the source sentence using a phrase-based statistical MT system, and a manually post-edited German sentence as a reference. In case of this dataset, the MT outputs are almost perfect and only little effort was required to post-edit the sentences. The results are evaluated using the human-targeted error rate (HTER) (Snover et al., 2006) and BLEU (Papineni et al., 2002). An important methodological point in the evaluation is that the systems are not evaluated with respect to original reference translation but using sentences that were created by manual post-editing of the MT system outputs.

To make the network focused more on editing the input sentence instead of preserving the meaning of the sentences, we represented the target sentence as a minimum-length sequence of edit operations needed to turn the machine-translated sentence into the reference post-edit. We extended the vocabulary by two special tokens *keep* and *delete* and then encoded the reference sentence as a sequence of *keep*, *delete* and *insert* operations with the insert operations by the actual words from the vocabulary. See Figure 5.4 for an example.

The decoder is a GRU network with 300 hidden units. Unlike in the MMT setup, we do not use the conditional GRU because it is prone to overfitting on the small dataset we work with.

The models were able to slightly, but statistically significantly improve over the baseline—leaving the MT output as is. The differences between the attention combination strategies are not significant.

5.1.4 Other Uses of the Attention Combination Strategies

Since publishing the attention combination strategies for RNN sequence-to-sequence models (Libovický and Helcl, 2017), our architectures have been used in several other publications.

Bawden et al. (2018) used the hierarchical attention model in MT experiments in which one previous sentence is used as additional input to the decoder. Hierarchical attention appeared to be the most successful method, improving the translation quality by around 1 BLEU point in all text genres that were evaluated.

Currey and Heafield (2018) experimented with adding explicit syntactic information into neural MT models. They use a linearized dependency parse of the source sentence and its delexicalized version as additional inputs to the model. All the inputs are processed with an independent RNN encoder and combined using the hierarchical attention model. In this way, they are able to improve the English-to-German translation, mostly for longer sentences, for which the translation quality usually degrades.

A paper introducing a Multimodal Summarization Task (Li et al., 2018) uses the hierarchical attention model as one of their baselines. The goal of the task is to generate a summary, i.e., the newspaper headline, of a short newspaper article and a photograph that illustrates the news story. Li et al. (2018) also introduced a more complex architecture that contains hierarchical attention as a subcomponent and achieves substantially higher score in terms of ROUGE (Lin, 2004).

Wang et al. (2018b) used hierarchical attention in end-to-end speech recognition utilizing multiple microphones located in different places. Hierarchical attention allows the decoder to decide what input is currently the most relevant for decoding.

Hierarchical attention was also used in the winning submission (**?**) to the Audio Visual Scene-Aware Dialog task at Dialog System Technology Challenge 7 (Yoshino et al., 2018). The task can be described as conversational question answering where an agent is supposed to answer a question about a visual scene, but before providing the final answer, it is can collect more information from the user. Hierarchical attention is used together with co-attention (Lu et al., 2016) to combine information from the visual scene and the conversational history.

5.2 Attention Models for Modality Combinations in Self-Attentive Sequence-to-Sequence Learning

*This section is based on the paper "*Input Combination Strategies for Multi-Source Transformer Decoder*", joint work with Jindřich Helcl and David Mareček, published as a research paper at WMT18.*

In 2017, the fully self-attentive Transformer model (Vaswani et al., 2017) became the new state of the art in MT, significantly outperforming the RNN-based models (Bahdanau et al., 2014) and increasing the training speed.

As already discussed in Section 2.3.3, the outline of the Transformer architecture is similar to the RNN-based models. The attention to the encoder is called cross-attention and is implemented as a separate sub-layer. The previously introduced methods for attention combination cannot be directly applied in the multi-head setup. We thus introduce four new attention combination strategies suited for the Transformer model. Two of them are a direct extension of the Transformer decoder with the cross-attention sub-layers connected either after each other or in parallel. The other two strategies are an adjustment of the attention strategies introduced for RNN models.

In the Transformer decoder, the cross attention is a separate sub-layer that follows the self-attentive sub-layer attending to the previously decoded symbols and before a feed-forward sub-layer. All the sub-layers are interconnected with residual connections. Formally, for a set of queries $\mathbf{Q}$, i.e., the decoder states, and the set of values $\mathbf{V}$, i.e., the encoder states, the multi-headed scaled dot-product attention with h heads is defined as:

$$\begin{aligned} \mathrm{Multihead}(\mathbf{Q}, \mathbf{V}) &= (\mathbf{H}_1 \oplus \cdots \oplus \mathbf{H}_h)\mathbf{W}^O && (5.12) \\ \mathbf{H}_i &= \mathrm{Attn}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{V}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) && (5.13) \\ \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} && (5.14) \end{aligned}$$

where $\mathbf{W}^O \in \mathbb{R}^{hd \times d}$, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^V \in \mathbb{R}^{d \times d}$ are trainable parameters, d is the dimension of the model. More details on the multi-headed scaled dot-product attention are provided in Section 2.3.2.

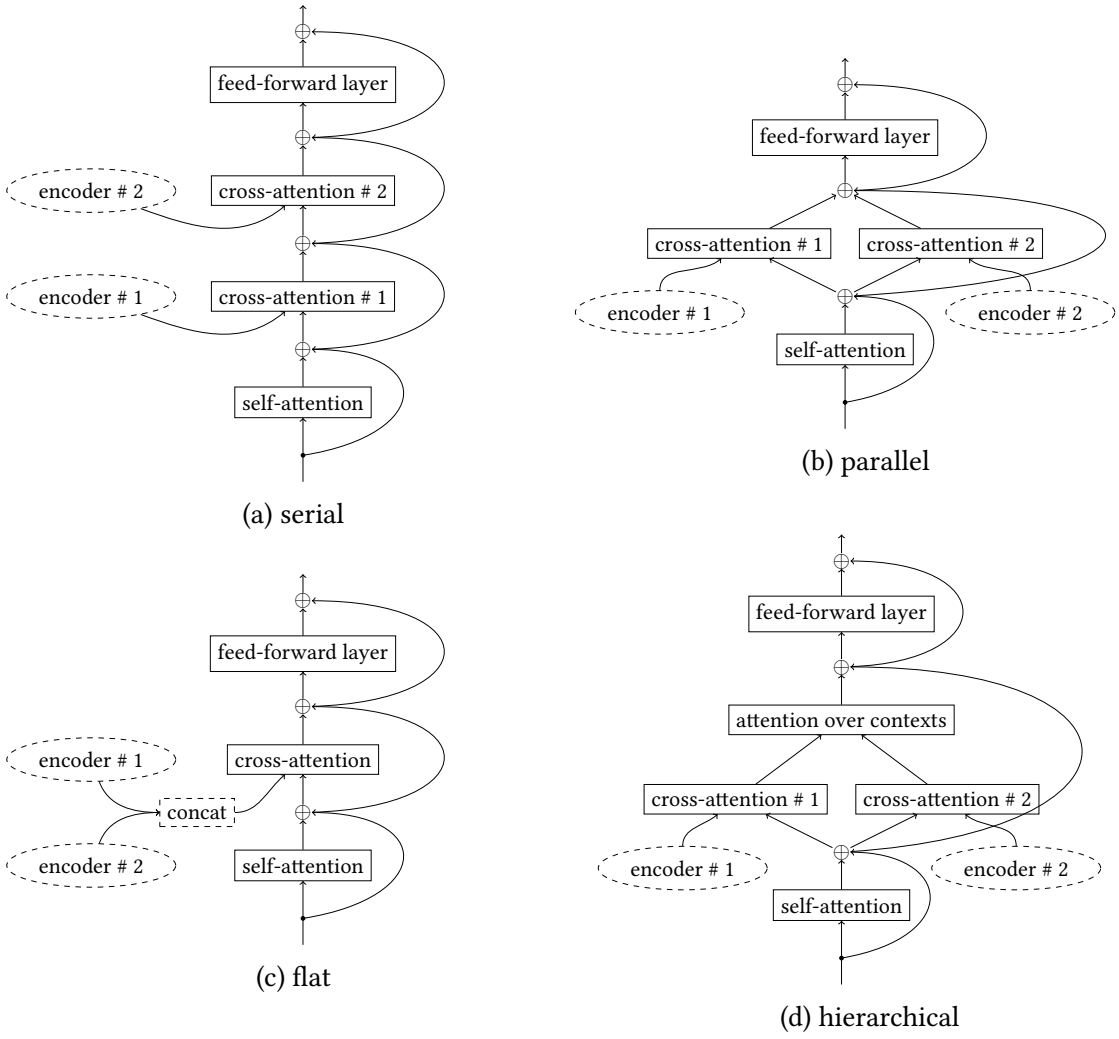


Figure 5.4: Schemes of computational steps of the serial, parallel, flat, and hierarchical attention combination in a single layer of the Transformer decoder.

5.2.1 Proposed Strategies

The input combination strategies are schematically depicted in Figure 5.4. A detailed description of the strategies follows. For clarity, we do not follow the upper-index notation from the previous section and denote values of the inputs $\mathbf{V}_i$ where i is the index of the input which is not consistent with the notation in the previous section where the lower index denotes position in a sequence of states.

Serial. In the serial strategy (Figure 5.4a), we compute the cross-attention in subsequent sub-layers for each input. The query set $\mathbf{Q}$ of the first cross-attention sub-layer is hence the set of the context vectors computed by the preceding self-attention sub-layer. The query set of each following cross-attention sub-layer is an output of the preceding sub-layer. All the sub-layers are interconnected with residual connections.

It can be recursively described:

$$\begin{aligned} \text{Multihead}_{\textit{serial}}(\mathbf{Q}, \mathbf{V}_{1:n}) &= \text{Multihead}(\text{Multihead}_{\textit{serial}}(\mathbf{Q}, \mathbf{V}_{1:n-1}), \mathbf{V}_n) + \mathbf{Q} \\ \text{Multihead}_{\textit{serial}}(\mathbf{Q}, \varnothing) &= \mathbf{Q} \end{aligned}$$

Parallel. In the parallel combination strategy (Figure 5.4b), the model attends to each encoder independently and then sums up the context vectors and form only one sub-layer in the decoder. Each input $\mathbf{V}_i$ is attended using the same set of queries $\mathbf{Q}$, i.e., the output of the self-attention sub-layer. The residual connection is used between the queries and the summed context vectors from the parallel attention.

$$\text{Multihead}_{\textit{parallel}}(\mathbf{Q}, \mathbf{V}_{1:n}) = \sum_{i=1}^{n} \text{Multihead}(\mathbf{Q}, \mathbf{V}_i) + \mathbf{Q} \tag{5.15}$$

Flat. The cross-attention in the flat combination strategy (Figure 5.4c) treats all the input states as a single set of keys and values. The cross-attention hence models a joint distribution over a flattened set of all input states:

$$\text{Multihead}_{\textit{flat}}(\mathbf{Q}, \mathbf{V}_{1:n}) = \text{Multihead}(\mathbf{Q}, (\mathbf{V}_1 \oplus \ldots \oplus \mathbf{V}_n)) + \mathbf{Q} \tag{5.16}$$

Unlike the approach that we have taken with the RNN models (Section 5.1, Equation 5.8), where the flat combination strategy requires an explicit projection of the input states into a shared vector space, in the Transformer models, representations on all layers are tied with residual connections. Therefore, the intermediate projection of the states of each encoder is not necessary, and we assume that we can share the projections within the attention heads.

Hierarchical. In the hierarchical combination (Figure 5.4d), we first compute the attention independently over each input. The resulting contexts are then treated as states of another input and the attention is computed once again over these states. Note that the query set Q is the same for all the attention computations.

$$\text{Multihead}_{hier}\left(\mathbf{Q}, \mathbf{V}_{1:n}\right) = \text{Multihead}(\mathbf{Q}, \text{concat}_{i=1\ldots n}(\text{Multihead}(\mathbf{Q}, \mathbf{V}_i)))$$

To be able to interpret the hierarchical attention in the multi-head setup, we do not compute the hierarchy within individual heads, but only after the outputs of the heads are projected to a single context vector. This projection also replaces the additional projection when computing the distribution β in the RNN model (Equation 5.10). It also theoretically allows utilizing a different number of heads for each input.

5.2.2 Experiments with Multimodal Translation

As in the case of the RNN models, we use Neural Monkey (Helcl and Libovický, 2017b; Helcl et al., 2018a)[3] for design, training, and evaluation of the experiments.

In all experiments, the encoder part of the network follows the Transformer BIG architecture (Vaswani et al., 2017, Table 3), including most of the hyperparameters. We use 6 layers in both the text encoder and decoder with the model dimension of 512. We set the dimension of the hidden layers in the feed-forward sub-layers to 4,096. The attention uses 16 heads.

The model is optimized with the Adam optimizer (Kingma and Ba, 2014) with initial learning rate 0.2, and Noam learning rate decay (Vaswani et al., 2017, Equation 3) with β_1 = 0.9, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, and 4,000 warm-up steps. The main difference from the original Transformer training setup is a smaller size of a mini-batch which we set to 32. During the decoding, we use beam search of width 10 and length normalization of 1.0 (Wu et al., 2016).

As in the previous experiments, we use the same image features as in the RNN setup, extracted from the last convolutional layer of the ResNet network (He et al., 2016). We apply a linear projection into 512 dimensions to adjust the image features to the model dimension. For each language pair, we create a shared wordpiece-based vocabulary of approximately 40 thousand wordpieces. We share the embedding matrices between the encoder and the decoder and use the transposed embedding matrix as the output projection matrix (Press and Wolf, 2017).

We evaluate the experiments the same way as the experiments with RNN architectures (Section 5.1.2), including the adversarial evaluation.

[3] `http://github.com/ufal/neuralmonkey`

English → *German*	BLEU	METEOR	Adversarial	
			BLEU	METEOR
RNN text-only	36.7 ± .8	55.1 ± .7	—	—
Transformer text-only	38.3 ± .8	56.7 ± .7	—	—
serial	38.7 ± .9	57.2 ± .6	37.3 ± .6	56.4 ± .7
parallel	38.6 ± .9	57.4 ± .7	38.2 ± .8	56.7 ± .7
flat	37.1 ± .8	56.5 ± .6	35.7 ± .8	54.7 ± .6
hierarchical	38.5 ± .8	56.5 ± .6	38.1 ± .8	56.5 ± .6

English → *French*	BLEU	METEOR	Adversarial	
			BLEU	METEOR
RNN text-only	48.3 ± .8	69.5 ± .6	—	—
Transformer text-only	59.6 ± .9	72.7 ± .7	—	—
serial	60.8 ± .9	75.1 ± .6	58.9 ± .9	73.7 ± .6
parallel	60.2 ± .9	74.9 ± .6	58.9 ± .9	74.0 ± .6
flat	58.0 ± .9	73.3 ± .7	57.0 ± .9	72.1 ± .6
hierarchical	60.8 ± .9	75.1 ± .6	60.2 ± .9	74.5 ± .6

English → *Czech*	BLEU	METEOR	Adversarial	
			BLEU	METEOR
RNN text-only	30.0 ± .8	29.1 ± .4	—	—
Transformer text-only	30.9 ± .8	29.5 ± .4	—	—
serial	31.0 ± .8	29.9 ± .4	29.7 ± .8	29.2 ± .4
parallel	31.1 ± .9	30.0 ± .4	30.4 ± .8	29.3 ± .4
flat	29.9 ± .8	29.0 ± .4	28.2 ± .8	28.2 ± .4
hierarchical	31.3 ± .9	30.0 ± .4	31.0 ± .8	29.7 ± .4

Table 5.3: Quantitative results of the MMT experiments with input combination strategies for the Transformer decoder on the 2016 test in terms of BLEU and METEOR.

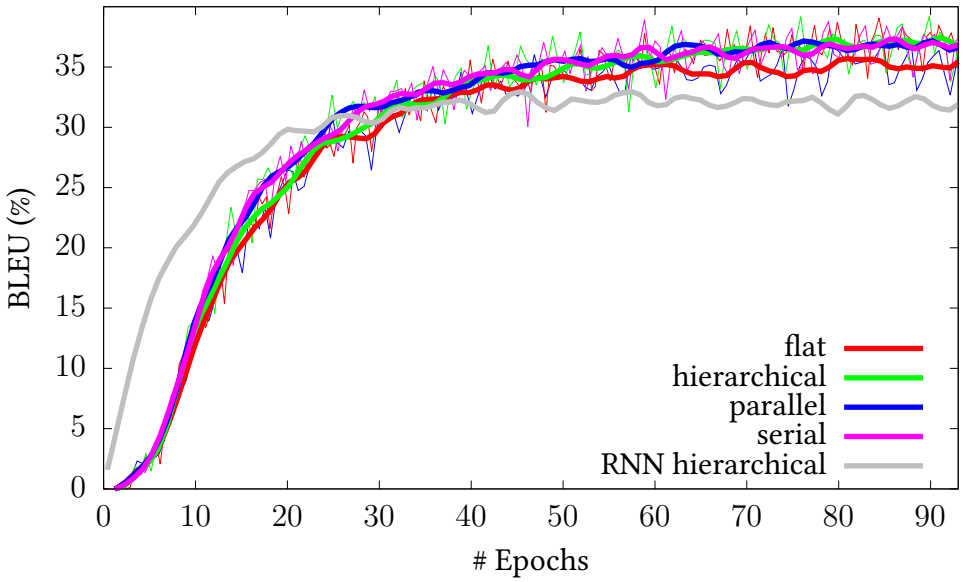


Figure 5.5: Learning curves of German Transformer-based MMT on validation data for context vector flat (red), hierarchical (green), parallel (blue) and serial (magenta) attention combination strategy compared to the hierarchical attention with RNN models (gray). The learning curves are smoothed the same way in Figure 5.1.

The quantitative results are tabulated in Table 5.3. In all cases, the models performed significantly better than RNN models. Unlike the experiments with the RNN models, the translation quality of the multimodal models is slightly better than the text-only baseline. However, the difference is statistically significant only in the case of the English-to-French translation.

The only case when the multimodal models score worse than the text-only models was the flat combination strategy. We hypothesize this might be because the optimization failed to find a common representation of the input modalities that could be used to compute the joint distribution, presumably because the image regions represented by the convolutional maps are not one-by-one semantically comparable with the wordpieces from the textual input.

The adversarial evaluation with randomly selected input images shows that all our models rely on both inputs while generating the target sentence and that providing incorrect visual input harms the translation quality. However, the fact that the scores are usually around 1 BLEU point lower, shows that the models rely almost exclusively on the textual part of the input than the image. The explicit modality choice in the hierarchical attention combination seems to make the models more robust to noisy visual input.

Learning curves from the experiments are displayed in Figure 5.5. Unlike in the strategies for the RNN models, there are basically no differences in convergence speed of the model except for the flat strategy that eventually reaches a worse translation quality. Compared to the hierarchical RNN model, the Transformer models converge more slowly in terms of the number of processed sentences. However, the total training time is smaller with the Transformer models which can be trained non-autoregressively.

5.2.3 Experiments with Multi-Source Machine Translation

In this set of experiments, we attempt to generate a sentence in a target language, given a set of equivalent sentences in multiple source languages.

To our knowledge, multi-source MT was previously studied only using the RNN-based models. Dabre et al. (2017) use concatenation of source sentences in various languages and process them with a single multilingual encoder. Zoph and Knight (2016) encode every sentence with a separate encoder and combine the inputs after independently computing context vectors with the attention mechanism. The techniques they propose are context vector concatenation and a gated sum of the context vectors. In all their experiments, the multi-source methods outperform the single-source baseline. Nishimura et al. (2018) deal with the setups when one of the source languages might be missing. Their system can benefit from having multiple sources but does not overly rely on a single source language which is the case of the models presented in this thesis.

We conduct our experiments using the Europarl corpus (Koehn, 2005). The Europarl corpus is compiled from the proceedings of the European Parliament which are published in all official languages of the European Union. The corpus contains sentences from the proceedings published between 1996 and 2011. This means that for some languages, only a subset of the corpus is available because of the respective countries joining the European Union during the period when the dataset was collected. The corpus is divided into bilingual sub-corpora with English as a pivot language.

We use Spanish, French, German, and English as source languages and Czech as a target language. We selected an intersection of the bilingual sub-corpora. Our dataset contains 511,600 5-tuples of sentences for training, 1,000 for validation and another 1,000 for testing.

We use almost the same setup as for the MMT experiments. Due to the memory demands of having four encoders, we use a smaller model than in the previous experiments. The encoders only consist of 4 layers and the decoder has 6 layers with embedding size 256, feed-forward layer dimension 2,048, and 8 attention heads. We

	MSMT		Adversarial evaluation (BLEU)			
	BLEU	METEOR	en	de	fr	es
baseline	16.5 ± .5	20.5 ± .3	—	—	—	—
serial	20.5 ± .6	23.5 ± .5	8.1 ± .4	19.7 ± .5	19.5 ± .6	18.4 ± .5
parallel	20.5 ± .6	23.3 ± .3	1.4 ± .2	18.7 ± .5	17.9 ± .5	20.3 ± .5
flat	20.4 ± .6	23.3 ± .3	0.2 ± .1	19.9 ± .6	20.0 ± .6	19.6 ± .5
hierarchical	19.4 ± .5	22.7 ± .3	4.2 ± .3	18.3 ± .5	18.3 ± .5	15.3 ± .5

Table 5.4: Quantitative results of the experiments with multi-source translation. The adversarial evaluation shows the BLEU scores when sentences of one of the input languages were randomly shuffled.

use a shared wordpiece vocabulary of 48 thousand wordpieces and share the embeddings between all encoders and the decoder. As in the MMT experiments, the transposition of the embedding matrix is reused as the parameters of the output projection layer (Press and Wolf, 2017). Due to the memory limitations, we reduce the batch size to 24.

We use bilingual English-to-Czech translation as a single source baseline. The baseline uses a different vocabulary of 42 thousand subwords trained only on the Czech-English parallel data.

We follow the evaluation protocol for MMT and evaluate the models using the BLEU and METEOR score. Similar to the MMT, we also perform an adversarial evaluation. To evaluate the importance of the source languages for the translation quality, we randomize the order of source sentences in respective languages one by one and observe the effect it has on the translation quality.

The results are shown in Table 5.4. All the proposed strategies perform better than the single-source translation from English to Czech. The best-scoring strategy is the serial stacking of the attention sub-layers. This result is in agreement with the MMT experiments (Table 5.3). Unlike the MMT where the flat combination did not appear to be a suitable technique, here it performs on par with other methods.

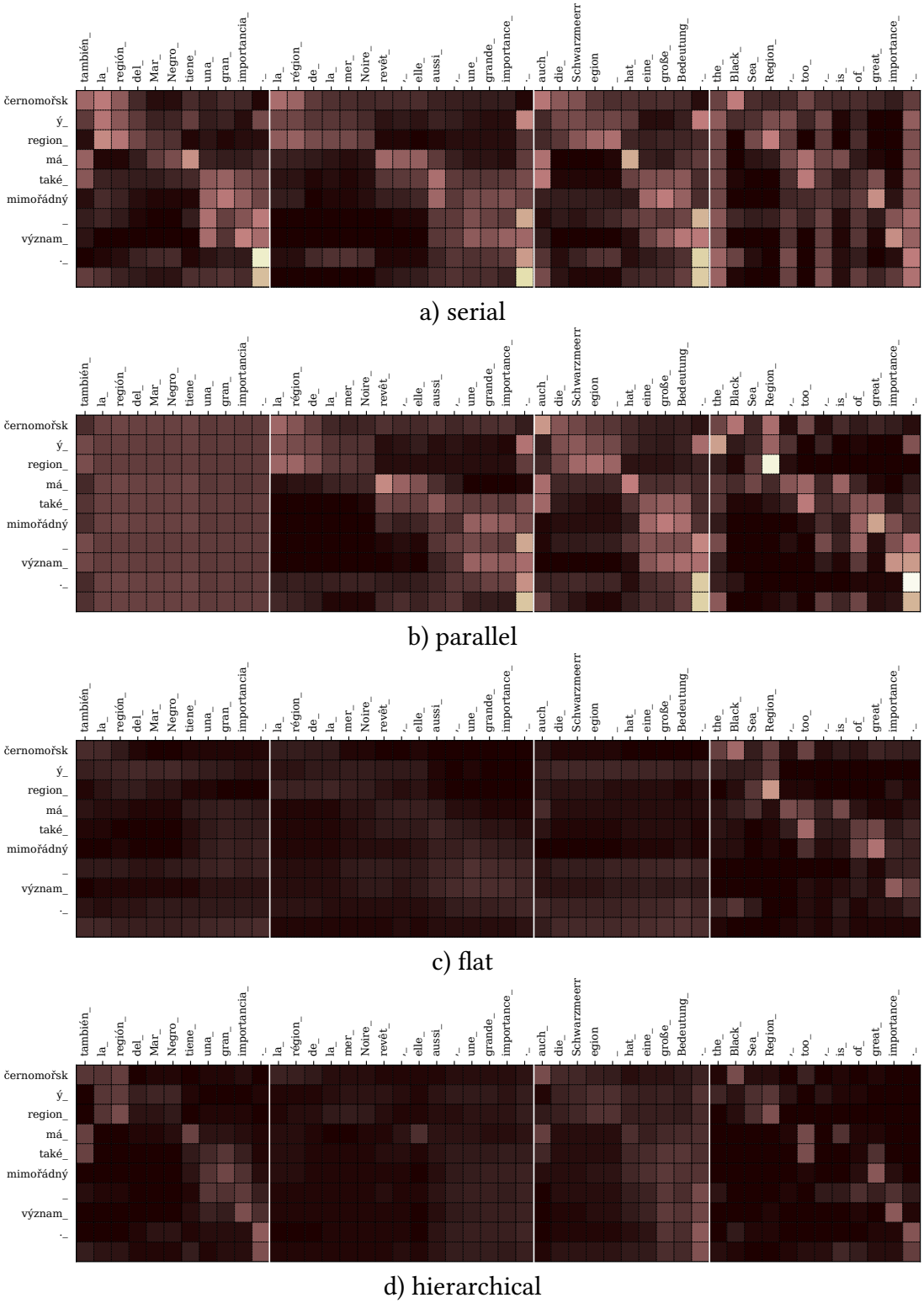
Figure 5.6 shows cross-attention weights visualizations for the four proposed combination strategies on Multi-source MT. The Czech target wordpieces are in rows, the source Spanish, French, German, and English wordpieces are concatenated and shown in columns. The attention values were taken from the fourth layer of the decoder and were averaged across the heads. For the serial and the parallel strategy, the cross-attention weights sum up to one for each language separately. The flat strat-

egy has only one common cross-attention over all input tokens. For the hierarchical strategy visualization, the cross-attention weights for the languages are multiplied by the weights from the attention distribution over the first context vectors. In the latter two cases, the complete rows sum up to one.

The visualization of the attention distributions shows that the serial strategy uses information from all source languages. The parallel strategy almost does not use the Spanish source. The flat strategy relies almost exclusively on English. The hierarchical strategy uses information from all source languages, however, the attention distributions seems to be fuzzier than in the other strategies.

The adversarial evaluation shows that all the models use English as the primary source. Providing incorrect English source always makes the translation incomprehensible. Introducing noise into other languages negatively affects the score on a much smaller scale. The usage of other languages seems to be arbitrary. Both the visualization in Figure 5.6 and Table 5.4 show that the hierarchical model relies also on the Spanish source sentences. On the other hand, in case of the parallel attention combination model, the output seems to be totally unaffected by Spanish.

The learning curves in Figure 5.7 show that the flat and hierarchical combination models converge from the beginning more slowly. A similar trend can be observed in the case of MMT, but the hierarchical combination at the end performs equally to other combination. The learning curve is steeper for the serial and the parallel combination strategies which eventually slightly outperform the others. The learning curve of the baseline bilingual model is at the beginning similarly steep as of the serial and parallel combinations but is eventually outperformed by all multilingual models.



a) serial

b) parallel

c) flat

d) hierarchical

Figure 5.6: Multi-source MT attention visualization. The rows correspond to the decoded wordpieces. The columns correspond to the source wordpieces. The source languages (Spanish, French, German and English) are separated by white lines. The visualizations were originally published by Libovický et al. (2018a)

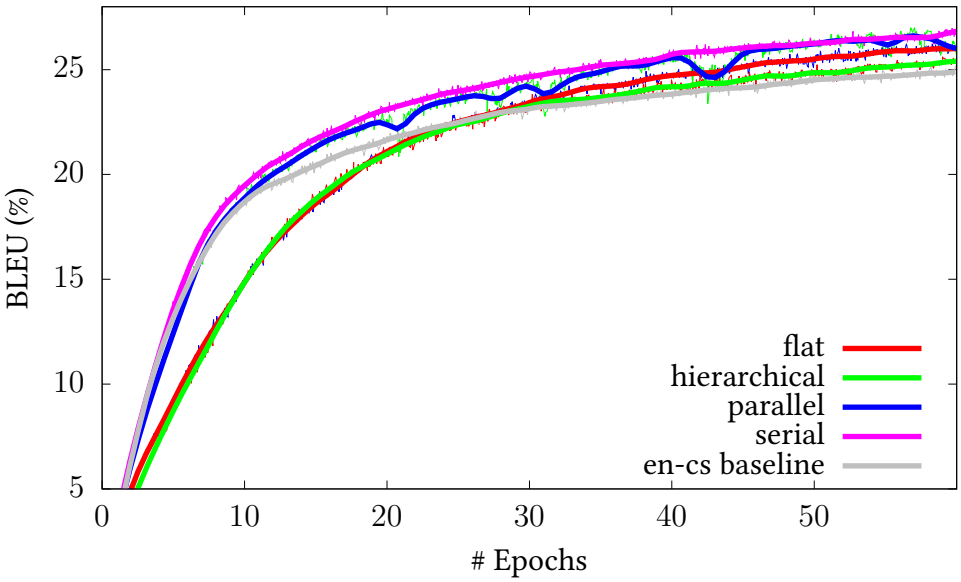


Figure 5.7: Learning curves for the multi-source MT on validation data for context vector flat (red), hierarchical (green), parallel (blue) and serial (magenta) attention combination strategy compared to the bilingual English-to-Czech baseline (gray). The learning curves are smoothed the same way as in Figure 5.1.

5.3 Improving Model Performance with Additional Data

*This section is based the papers "*CUNI Submission for the WMT17 Multimodal Translation Task*" and "*CUNI Submission for the WMT18 Multimodal Translation Task*", joint work with Jindřich Helcl and Dušan Variš published as system description papers at WMT17 and WMT18.*

To boost the translation quality in the shared task evaluation, we experimented with methods for increasing the volume of training data. In the first part of this section, we present methods of acquiring additional textual data by mining in-domain data from parallel corpora.

In the second part of this section, we experiment with the Imagination model (Elliott and Kádár, 2017) which allows adding monolingual image captioning data in the source language.

Following the terminology from the WMT MMT Task, models that use only the Multi30k data are called *constrained*, models utilizing also the additional data are called *unconstrained*.

5.3.1 Acquiring Additional Training Data

Neural MT requires a large amount of training data to work well. Koehn and Knowles (2017) claim that in order to outperform a phrase-based MT system in terms of BLEU, we need training data of over 10^7 words and more than 10^8 words to outperform a statistical system with a LM trained on large monolingual data. Although this empirical observation clearly does not hold for the Multi30k dataset with 29,000 training instances (Moses reaches 32.5 BLEU points, the Transformer gets 38.3 BLEU points, see Table 4.4), it is clear that acquiring additional training data has the potential to improve translation quality.

We already mentioned in Section 4.1 that image captions are a specific genre of text where the language is rather limited in terms of vocabulary and grammatical means. The sentences are almost always in the present tense, there are almost no named entities and the sentences rarely consist of more than one clause. On one hand, we would like to extend the training data as much as possible, on the other hand, we would like to avoid forcing the model to deal with linguistic phenomena that are not present in the Multi30k dataset.

To acquire additional parallel training data, we used a filtering technique to select in-domain sentences from both parallel and monolingual data similar to a method by Yasuda et al. (2008). We trained an RNN character-level LMs for the target languages using the sentences available in the training part of the Multi30k dataset. We used a GRU network with 512 hidden units and character embedding size of 128.

We used a LM to select sentence pairs from parallel corpora. By scoring the German part of several parallel corpora: EU Bookshop (Skadiņš et al., 2014), News Commentary (Tiedemann, 2012) and CommonCrawl (Smith et al., 2013). In this way, were only able to retrieve a few hundreds of in-domain sentences. For that reason, we also included sentences with a higher perplexity filtered using additional rules. We analyzed the sentences using spaCy[4] and used the following empirical rules for filtering: The sentences

- Must have between 2 and 30 tokens;
- Must be in the present tense;
- Must not contain non-standard punctuation, numbers of multiple digits, acronyms, or named entities; and
- Must have at most 15% out-of-vocabulary rate with respect to the vocabulary inferred from the training part of Multi30k.

[4] `https://spacy.io/`

We extracted additional 3,000 in-domain parallel sentences using these rules. Examples of the additional data are given in Table 5.5.

For Czech, we computed perplexities of the Czech part of the CzEng corpus (Bojar et al., 2016b). We selected 15 thousand low-perplexity sentence pairs out of 64 million sentence pairs in total by setting the perplexity threshold to 2.5.

By applying the same approach to the French versions of the corpora, we were able to extract only a few additional in-domain sentences. We thus only use the Multi30k as in-domain data.

For German, we selected 30,000 best-scoring German sentences from the monolingual The SDEWAC corpus (Faaß and Eckart, 2013) which were both semantically and structurally similar to the sentences in the Multi30k dataset. SDEWAC corpus consists of 880 million sentences crawled from the web. Following Calixto et al. (2017), we back-translated (Sennrich et al., 2016a) the German captions intended for cross-lingual captioning (i.e., 5 captions for each image), and sentences retrieved from the SDEWAC corpus. We include these back-translated sentence pairs as additional training data for the text-only systems. The back-translation system was a text-only RNN model that was trained on the Multi30k dataset only.

For Czech, we applied the perplexity criterion on monolingual corpora and used back-translation to create synthetic parallel data. We scored 333 million sentences from the CommonCrawl corpus and 66 million sentences from the News Crawl data (which is used in the WMT News Translation Task; Bojar et al., 2016a) and extracted 18 thousand and 11 thousand sentences from these datasets respectively.

Finally, we use the whole EU Bookshop corpus (Tiedemann, 2012) as an additional out-of-domain parallel data. Since the size of this dataset is large compared to the sizes of the other parts, we oversample the rest of the data to balance the in-domain and out-of-domain portions of the training dataset. The oversampling factors are shown in Table 5.6.

To summarize, we compiled an order of magnitude bigger training datasets for all the three language pairs which should increase the variability in the training data but at the same time retain the properties of the image captioning domain.

SDEWAC (with back-translation)	
German	*English*
Zwei Männer unterhalten sich	Two men are talking to each other.
Menschen auf der Straße.	People on the street.
Ein kleines Mädchen sitzt auf einer Schaukel.	A little girl is sitting on a swing.
Eine junge Frau sitzt auf einer Bank und liest ein Buch.	A young woman sits on a bench reading a book.
Eine Katze braucht Unterhaltung.	A cat is having a discussion.
Dieser Knabe streichelt das Schlagzeug.	This professional is petting the drums.
Parallel Corpora	
German	*English*
Menschen bei der Arbeit	People at work
Kinder und Jugendliche in der Stadt	Children and young people in the urban environment
Männer und Frauen	Men and women
Sicherheit bei der Arbeit	Safety at work
Personen in der Öffentlichkeit	Members of the public
Frauen und Männer	Women and men

Table 5.5: Random examples of the collected additional training data for English-to-German translation.

	de	fr	cs
Multi30k		29k	
oversampling factor	273×	366×	9×
Task 2 BT	145k	—	—
in-domain parallel	3k	—	15k
in-domain back-translation	30k	—	29k
oversampling factor	39×	—	7×
EU Bookshop	9.3M	10.6M	445k
MS COCO (English only)		414k	

Table 5.6: Overview of the data used for training our models with oversampling factors. The EU Bookshop data were not oversampled.

5.3.2 Imagination Model

Another way of enriching the training data that is more sophisticated than adding more parallel data was introduced with the Imagination model (Elliott and Kádár, 2017). This model employs multi-task learning (Caruana, 1997) when sharing the encoder for two different tasks: generating a sentence in the target language and predicting a representation of an image corresponding to the sentence. The model allows using not only additional parallel training data, but also monolingual image captioning data.

The imagination component serves effectively as a regularizer to the encoder, making it consider the visual aspects of meaning and presumably making the encoder representation semantically richer. This is achieved by training the model to predict the image representations that correspond to those computed by a pre-trained image classification network. Given a set of encoder states $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_{T_x}) \in \mathbb{R}^{T_x \times d_h}$, the model computes the predicted image representation as follows:

$$\hat{\mathbf{y}}_{img} = \mathbf{W}_2^R \, \mathrm{ReLu} \left(\mathbf{W}_1^R \sum_j \mathbf{h}_j + \mathbf{b}_1 \right) + \mathbf{b}_2 \tag{5.17}$$

where $\mathbf{W}_1^R \in \mathbb{R}^{d_h \times d_i}$, $\mathbf{b}_1 \in \mathbb{R}^{d_i}$, $\mathbf{W}_2^R \in \mathbb{R}^{d_i \times d_y}$, and $\mathbf{b}_2 \in \mathbb{R}^{d_y}$ are trainable parameter matrices, d_i is the dimension of the intermediate projection, and d_y is the image representation. Equation 5.17 corresponds to a single-hidden-layer feed-forward network with a Rectified Linear Unit (ReLU) activation function (Nair and Hinton, 2010), applied to the sum of the encoder states.

Since the encoder part of the model is shared, additional weight updates are propagated to the encoder during the model optimization with respect to this additional objective. For the generated image representation $\hat{\mathbf{y}}$ and the reference representation $\mathbf{y}$, the error is estimated as a margin-based loss with margin parameter $\alpha \in \mathbb{R}$:

$$L_{imag} = \max\left(0, \alpha + \mathrm{dist}(\hat{\mathbf{y}}, \mathbf{y}) - \mathrm{dist}(\hat{\mathbf{y}}, \mathbf{y}_c)\right) \tag{5.18}$$

where $\mathbf{y}_c$ is a contrastive example randomly drawn from the training mini-batch and dist is a distance function between the representation vectors, in our case the cosine distance, α is a hyperparameter. This loss function is frequently used in similar setups (Chrupała et al., 2017; Elliott and Kádár, 2017) because it empirically performs better than different regression loss functions.

Unlike Elliott and Kádár (2017), we sum both translation and imagination losses within the training batches rather than alternating between the training of each component separately.

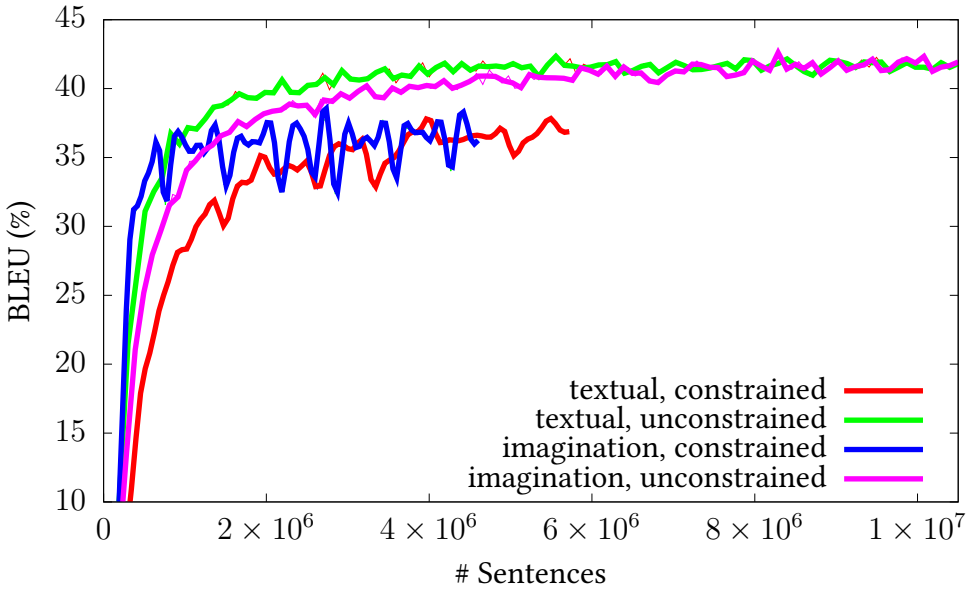


Figure 5.8: Learning curves for the MMT into German using the Transformer model. The plot compares the text-only model with the constrained (red) and unconstrained dataset (green) data and the Imagination models with the constrained (blue) and unconstrained (magenta) data. Note that the horizontal axis shows the number of processed sequences because the number of epochs differs among experiments.

5.3.3 Experiments

In the experiments with additional data and the Imagination model, we use the same RNN and Transformer architectures that were discussed in Sections 5.1 and 5.2 respectively.

In the RNN setup, it means a single layer bidirectional GRU encoder with 300 hidden units and embeddings of 300 dimensions. The decoder uses the conditional GRU (Firat and Cho, 2016) with 500 hidden units and a vocabulary of 20 thousand BPE tokens.

In the Transformer models, we use an encoder and a decoder with 6 layers, model dimension 512 and 4,096 units in the feed-forward layers. We used a wordpiece-based vocabulary of approximately 20 thousand symbols.

The results are tabulated in Table 5.7. With the Transformer model, adding the parallel data improved the translation quality by 1.5 BLEU points for each of the language pairs. An additional improvement was achieved by adding the Imagination component. The biggest gain was achieved in Czech where the Imagination model adds 4 BLEU points. Results of experiments with the RNN models are inconclusive. The translation quality increased for German but significantly dropped for French and Czech.

		English → German	BLEU	METEOR
RNN	Cons.	Textual	36.7 ±.8	55.1 ±.7
		Imagniation	36.8 ±.8	55.3 ±.6
		Multimodal (hierarchical)	37.6 ±.8	56.0 ±.6
	Unc.	Textual	38.7 ±.8	56.3 ±.7
		Imagination	38.2 ±.8	56.7 ±.7
Tranformer	Cons.	Textual	38.3 ±.8	56.0 ±.7
		Imagniation	39.2 ±.8	56.8 ±.7
		Multimodal (serial)	38.7 ±.8	57.2 ±.6
	Unc.	Textual	40.4 ±.9	59.0 ±.6
		Imagination	42.6 ±.8	59.4 ±.6

		English → French	BLEU	METEOR
RNN	Cons.	Textual	48.3 ±.8	69.5 ±.6
		Imagniation	47.6 ±.8	68.8 ±.6
		Multimodal (hierarchical)	48.2 ±.9	69.2 ±.6
	Unc.	Textual	42.8 ±.8	66.3 ±.6
		Imagination	43.4 ±.8	67.0 ±.6
Tranformer	Cons.	Textual	59.6 ±.9	74.1 ±.6
		Imagniation	59.7 ±.9	74.4 ±.6
		Multimodal (serial)	60.8 ±.9	75.1 ±.6
	Unc.	Textual	62.5 ±.8	76.7 ±.6
		Imagination	62.8 ±.9	77.0 ±.6

		English → Czech	BLEU	METEOR
RNN	Cons.	Textual	30.0 ±.8	29.1 ±.4
		Imagniation	29.8 ±.8	28.8 ±.4
		Multimodal (hierarchical)	29.5 ±.8	28.7 ±.4
	Unc.	Textual	27.7 ±.8	27.6 ±.4
		Imagination	31.0 ±.8	30.0 ±.4
Tranformer	Cons.	Textual	30.9 ±.8	29.5 ±.4
		Imagniation	30.5 ±.9	29.6 ±.4
		Multimodal (serial)	31.0 ±.8	29.9 ±.4
	Unc.	Textual	32.3 ±.9	30.7 ±.4
		Imagination	36.3 ±.9	32.8 ±.4

Table 5.7: Results on the 2016 test set in terms of BLEU score and METEOR score.

The learning curves in Figure 5.8 show a different training dynamics of the constrained and unconstrained Transformer models. The unconstrained models converge from the beginning of the training faster than the constrained models. The learning curve of the unconstrained Imagination model is remarkably close to the text-only model, even though the training data contains target sentences for only a half of the training instances. The Imagination model in the constrained setup has a steep learning curve at the beginning, however, the validation BLEU score becomes rather unstable as the model starts to overfit.

By the time, we achieved the presented results, it was the state-of-the-art translation quality on the Multi30k dataset. Recently, the new state of the art was achieved by Grönroos et al. (2018) with the Transformer architecture with the same hyperparameters as ours implemented in OpenNMT (Klein et al., 2018). As discussed in Section 4.3, their improvement comes from utilizing large parallel corpora, creating a synthetic dataset based on MS COCO and more careful preprocessing of the data.

6

Analysis of Multimodal Translation Systems

In the previous chapter, we have introduced several methods of combining multiple inputs in sequence-to-sequence models based on Recurrent Neural Networks (RNNs) (Section 5.1) and Self-Attentive Network (SAN) (Section 5.2). We further discussed techniques for improving translation quality by collecting additional training data (Section 5.3). We used automatic metrics for estimating the translation quality and adversarial evaluation to asses whether the models rely on the image information or not. However, we did not conduct any deeper analysis either of the translation outputs or the model themselves.

In this chapter, we go beyond the standard automatic evaluation metrics and try to asses the models in more detail. In Section 6.1, we examine how the translation quality is influenced by the objects present in the source image and by linguistic phenomena occurring in the source sentence. In Section 6.2, we intrinsically evaluate the representations learned by the Multimodal Machine Translation (MMT) models and compare them with other multimodal and monomodal models on sentence semantic similarity and image retrieval tasks.

6.1 Model Performance Analysis

In this section, we evaluate translation quality on the sentence level and try to assess how the properties of the image and the properties of the source sentences influence the translation quality.

object	frequency
person	896
dog	70
car	50
chair	38
bicycle	27
cup	13
backpack	11
handbag	9
bottle	8

Table 6.1: The most frequent objects detected in the test part of the Multi30k dataset and their frequency.

For all systems that we have introduced through this thesis (Sections 5.1, 5.2, and 5.3), we obtain the sentence-level BLEU score (Chen and Cherry, 2014) and compute Pearson correlation with quantitative properties of the source images and sentences.

We processed the images from the test set with the YOLO object detector (Redmon et al., 2016) and selected all objects that appear at least ten times in the test set. The YOLO object detector is a ResNet-based deep neural network which detects objects and classifies them into 80 categories used in the MS COCO dataset. The most frequently detected objects and their frequencies in the 1,000 images of the test set are tabulated in Table 6.1.

On the language side, we are interested in how the following features influence the translation quality:

- Source sentence length;
- Number/proportion of auxiliary and content verbs;
- Tense used in the source sentence;
- Average log-frequency of the source words in the training data;
- Presence of numbers (image captioning models tend to have problems with counting objects);
- Presence of words denoting colors.

The features were extracted by rules based on part-of-speech tags and dependency parsing obtained from spaCy[1].

[1] `https://spacy.io/`

The models that we evaluated were all RNN and Transformer models that were mentioned in the previous sections: multimodal models using various attention combination strategies, text-only models, and imagination models trained in both the constrained and unconstrained setup. The detailed results are tabulated in Table 6.2. We only show the table for German as a target language as the remaining two target languages deliver similar results.

All correlation values lie in the interval from -0.2 to +0.2 with only minor differences between the text-only and multimodal models. The most important conclusion that types of model and the choice of the target language does not influence how the translation quality depends on the features of the input images and source sentences. Our main observations are that, first, the unconstrained models seem to be more independent on the measured input qualities and, second, that the RNN-based models are more sensitive to all tracked features.

Presumably, the more complex the source sentence is, the worse translation quality we obtain. Longer sentences, sentences with more nouns and content verbs tend to score worse. With the increasing number of objects detected in the images, the sentence-level BLUE drops as well.

Sentences in past tense seem to score worse. This might be either because of underfitting due to the low number of past sentences in the training data, or the variability in how the past tense might be expressed in the target languages (preterite or perfective in German) which might differ from the reference sentence. The translation quality seems to be positively affected by the word frequency in the training data and the presence of words expressing colors.

The sentence-level BLEU score appears to be mostly independent of what objects are in the input image. System outputs for images with people tend to receive lower scores, whereas sentences outputs for images with dogs tend to receive higher scores. We believe that this can be explained by low source sentence variability in the respective object categories.

	RNN	Text-only		Imagination		Multimodal			
		con.	unc.	con.	unc.	init.	concat	flat	hier.
object count	>90% confidence	-.14	-.13	-.12	-.16	-.13	-.16	-.17	-.14
	>50% confidence	-.14	-.13	-.14	-.16	-.14	-.17	-.17	-.13
objects in the image	person	-.10	-.06	-.08	-.10	-.08	-.11	-.10	-.11
	car	.04	.05	.05	.06	.05	.05	.05	.07
	chair	-.03	.01	.00	-.05	-.03	-.04	-.04	-.03
	dog	.08	.04	.09	.09	.08	.09	.09	.09
	handbag	.01	.03	.01	.02	.05	.02	.00	.02
	bicycle	-.04	-.05	-.05	-.06	-.08	-.05	-.06	-.06
source sentence properties	sentence length	-.10	-.12	-.11	-.13	-.12	-.13	-.10	-.09
	non-aux. verb ratio	-.13	-.12	-.11	-.11	-.12	-.11	-.13	-.12
	aux. verb ratio	.13	.15	.16	.15	.17	.14	.19	.16
	noun count	-.13	-.15	-.13	-.15	-.14	-.15	-.13	-.13
	past tense	-.15	-.15	-.16	-.14	-.15	-.17	-.15	-.14
	contains numeral	.00	-.03	-.02	-.02	-.02	-.02	-.01	-.03
	contains color	.15	.12	.14	.11	.15	.14	.19	.19
	avg. word freq.	.27	.26	.28	.29	.29	.29	.27	.28

	Tranformer	Text-only		Imagination		Multimodal			
		con.	unc.	con.	unc.	flat	hier.	par.	serial
object count	>90% confidence	-.11	-.05	-.08	-.08	-.09	-.11	-.11	-.11
	>50% confidence	-.11	-.06	-.09	-.09	-.10	-.11	-.10	-.12
objects in the image	person	-.08	-.03	-.07	-.05	-.06	-.09	-.08	-.07
	car	.02	.01	.03	-.01	.03	.01	.00	.00
	chair	.04	.01	.04	.01	.01	.03	.03	.01
	dog	.04	.02	.05	.03	.06	.06	.06	.05
	handbag	-.01	-.01	-.01	-.01	.00	-.01	-.01	.00
	bicycle	-.02	-.01	-.02	-.02	-.02	-.05	-.02	-.03
source sentence properties	sentence length	-.02	.05	.01	.02	-.02	-.01	.01	.02
	non-aux. verb ratio	.02	.00	.03	.00	-.01	-.01	-.01	-.02
	aux. verb ratio	.10	.05	.07	.09	.13	.12	.07	.11
	noun count	-.15	-.05	-.10	-.07	-.11	-.11	-.10	-.09
	past tense	-.08	-.04	-.07	-.06	-.11	-.09	-.08	-.11
	contains numeral	.00	.04	.01	.01	-.03	-.01	-.03	-.01
	contains color	.17	.09	.15	.13	.18	.18	.17	.17
	avg. word freq.	.22	.13	.17	.14	.22	.23	.21	.19

Table 6.2: Pearson correlation of sentence-level BLEU with quantitative properties of source sentences and images for the test part of Multi30k data for translation from English to German. For text-only and imagination models, we report both the constrained ('con.') and unconstrained ('unc.') data setup. With the multimodal models, we only use the Multi30k dataset and report all variants of adding the visual information introduced in Sections 5.1 and 5.2.

6.2 Assessing Representations Learned by the Models

*This section is based on the paper "*Assessing Representations Learned in the Multimodal Translation Models*", joint work with Pranava Madhyastha, currently under review for ACL 2019.*

In the previous section, we have shown that multimodal information can improve the quality of the Machine Translation (MT) systems. Experiments with representation learning (see Section 3.2) suggest that the improved translation quality might be a consequence of improved quality of the representation learned in the networks. There are other experiments showing the usefulness of grounding representation learned in deep learning models by conditioning on multimodal information (Chrupała et al., 2015).

On the other hand, recent work towards universal sentence representation shows that language modeling on large-scale datasets can provide sufficiently informative representations reusable in most Natural Language Processing (NLP) tasks while reaching state-of-the-art results in most of them (Peters et al., 2018; Devlin et al., 2018).

In this section, we systematically approach these seemingly contradictory results and investigate representations obtained specifically from grounded models using RNNs and SANs. Our main observations are:

- Models with explicit access to visual information learn to ignore image information;
- Grounding representation in *visual modality leads to semantically better representation* as it provides a stronger training signal and is especially pronounced when the model has access to less training samples;
- While Transformer-based models might have better task performance, we observe that *RNN-based models capture semantic information better* .

In the rest of this section, we intrinsically evaluate all MMT models presented in the previous sections and compare them with Language Models (LMs) and image retrieval models with similar architectures. We obtain a highly correlated mutual projection between representations of source sentences and representations of the corresponding images, and we evaluate the ability to capture visual features. By assessing the representations on the Semantic Textual Similarity (STS) task, we evaluate their semantic properties. In addition, we compute the Distance Correlation (DC) between representations from all pairs of the probed models.

6.2.1 Related Work

Previous research proposed several ways to evaluate representations that emerge in neural models. The most common technique is measuring correlation between similarity of learned representations and semantic similarity of words (Hill et al., 2015; Gerz et al., 2016) and sentences (Agirre et al., 2012, 2016). Other methods include relating the representations to the existing well-trained models by finding mutual projection between the representation in the probed model and observing how the projected representations perform within the trained model (Saphra and Lopez, 2018) or observing the effect of changes in the representation by backpropagating the changes to the input (Poerner et al., 2018).

Universal sentence representations are often evaluated on downstream tasks. Conneau and Kiela (2018) and Wang et al. (2018a) recently introduced comprehensive sets of such downstream tasks providing a benchmark for the sentence representation evaluation. The tasks include various sentence classification tasks, entailment or coreference resolution. The drawback of these methods is that they require generating representations of millions of sentences which are later used for a rather time-consuming training of models for the downstream tasks.

6.2.2 Assessing Contextual Representations

In this section, we describe how we extract the representations and the methods we use for the representation evaluation.

Contextual Representation. By contextual representation, we mean hidden states of a network processing textual input, in our case either an RNN or a SAN. Unlike the embedding layer which assigns vectors to all tokens independently, hidden states on later layers already may contain information from the rest of the sentence, therefore we call the representations contextual.

In all our models, the number of hidden states is the same as the number of input tokens. Because all the evaluation methods that we use, require a fixed-sized sentence representation, we mean-pool the hidden states, i.e., compute the arithmetic mean over the sentence length. In the rest of this section, we call this vector a representation of a sentence.

Canonical Correlation Analysis. We use Canonical Correlation Analysis (CCA) over the mean-pooled sentence representations and image representations to obtain two highly correlated projections respectively. CCA and its variants have been used in previous research to obtain cross-modal representations (Gong et al., 2014; Yan and Mikolajczyk, 2015).

We take input as the two sets of aligned representations from two different subspaces, say $\mathbf{T} = \{\mathbf{t_1}, \ldots, \mathbf{t_n}\}$ and $\mathbf{V} = \{\mathbf{v_1}, \ldots, \mathbf{v_n}\}$, where $\mathbf{t_i}$ and $\mathbf{v_i}$ are vector representations. CCA (Hotelling, 1936) finds pairs of directions $\mathbf{W_t}, \mathbf{W_v}$, such that the linear projections of $\mathbf{T}$ and $\mathbf{V}$ onto these directions, i.e., the canonical representations $\mathbf{W_t}^\top \mathbf{T}$ and $\mathbf{W_v}^\top \mathbf{V}$, are maximally correlated:

$$\mathbf{W}_t, \mathbf{W}_v = \underset{\mathbf{W}'_t, \mathbf{W}'_v}{\operatorname{argmax}} \operatorname{corr}\left(\mathbf{W_t}^\top \mathbf{T}, \mathbf{W_v}^\top \mathbf{V}\right) \tag{6.1}$$

For further details on CCA, we refer the reader to Hardoon et al. (2004).

The most significant property of CCA for our analysis is that CCA is a *subspace only method* where we only obtain a highly correlated linear transformation for each of the representations given paired sentence and image representations. We do not explicitly learn any new information. We evaluate the projected representations on image retrieval task and report Recall at 10, i.e., a proportion of cases when the correct image is within the 10 nearest neighbors of the sentence representation.

We fit the CCA on the 29,000 image-sentence pairs of the training portion of the Multi30k and evaluate on the 1,000 pairs from the test set.

Cosine Distance. Besides the ability to represent the visual content of an image, we evaluate the representation on the STS task. The STS task focuses on evaluation of semantic similarity of sentences regardless of their relation to the visual modality and thus can capture more abstract aspects of meaning that cannot be represented visually in a straightforward manner.

For the STS task, we use cosine distance between vectors $\mathbf{t}$and $\mathbf{v}$:

$$\operatorname{sim}(\mathbf{t}, \mathbf{v}) = 1 - \frac{\mathbf{t} \cdot \mathbf{v}}{\|\mathbf{t}\|\|\mathbf{v}\|}. \tag{6.2}$$

Following the SentEval benchmark (Conneau and Kiela, 2018), we report the Spearman correlation between the cosine distance and human assessments. We evaluate the representations on the STS SemEval 2016 task (Agirre et al., 2016). The test set consists of 1,186 sentence pairs collected from datasets of newspaper headlines, machine translation post-editing, plagiarism detection, and question-to-question and answer-to-answer matching on the Stack Exchange data. Each sentence pair is annotated with a similarity value. Similar to the image retrieval task, we do not fine-tune the representations for the task.

Distance Correlation. The third method that we use for probing the representations learned in the MMT and other models is measuring a mutual similarity of representations. We measure the similarity using DC.

DC is a measure of dependence between any two paired vectors of arbitrary dimensions (Székely et al., 2007). Given, two paired vectors, $\mathbf{t} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$ and suppose that $\phi_1(\mathbf{t}), \phi_2(\mathbf{v})$ and $\phi_3(\mathbf{t}, \mathbf{v})$ are the individual characteristic functions and joint characteristic function of the two vectors respectively. The distance covariance between $\mathbf{t}$ and $\mathbf{v}$ with finite first moments is a non-negative number given by:

$$\text{dcov}^2(\mathbf{t}, \mathbf{v}) = \int_{\mathbb{R}^{m+n}} \|\phi_3(\mathbf{t}, \mathbf{v}) - \phi_1(\mathbf{t})\phi_2(\mathbf{v})\|_2^2 \, \psi(\mathbf{t}, \mathbf{v}) d\mathbf{t} \, d\mathbf{v}, \tag{6.3}$$

where $\psi(\mathbf{t}, \mathbf{v}) = \{\|\mathbf{t}\|_m^{1+m} \|\mathbf{v}\|_n^{1+n}\}^{-1}$; m and n are the dimensionalities of $\mathbf{t}$ and $\mathbf{v}$ respectively. The DC is then defined as:

$$\text{dcorr}(\mathbf{t}, \mathbf{v}) = \frac{\text{dcov}(\mathbf{t}, \mathbf{v})}{\sqrt{\text{dcov}(\mathbf{t}, \mathbf{t})\text{dcov}(\mathbf{v}, \mathbf{v})}} \tag{6.4}$$

A detailed description of DC is beyond the scope of this thesis, but we refer the reader to Székely et al. (2007) for a thorough analysis.

Our use of DC is motivated by the result that DC quantifies dependence measure, especially it equals zero exactly when the two vectors are mutually independent and are not correlated.

Also, DC measures both *linear* and *non-linear* association between two vectors, unlike the CCA and cosine similarity, which are linear. We are especially interested in studying the degree to which two independently learned representations are correlated.

6.2.3 Experiments

In our experiments, we assess representations learned in all MMT models presented in Chapter 5. We compare their representations with representations from LMs and image retrieval models with comparable architectures trained on dataset of different sizes.

Datasets

For all models, we use the training part of Multi30k (Elliott et al., 2016) that consists of only 29 thousand training images with English captions and their translations. For monolingual experiments, we further use English captions from the Flickr30k dataset (Plummer et al., 2015) with 5 captions for each image, in total 145 thousand sentences. We also use the MS COCO data (Lin et al., 2014), with 414 thousand descriptions of 82 thousand images. For more details on the multimodal datasets, see Sections 3.3 and 4.2.

We also included the unconstrained MMT, with additional data harvested from parallel and monolingual corpora (Helcl and Libovický, 2017a; Helcl et al., 2018b) combined with the EU Bookshop corpus (Tiedemann, 2012), in total of 200 million words. See Section 5.3 for more details.

Models

We evaluate the following MMT models:

- RNN text-only models trained on Multi30k (Section 5.1) and on the unconstrained data (Section 5.3)
- RNN multimodal models with concatenation, flat and hierarchical attention combination strategy trained on Multi30k (Section 5.1)
- RNN Imagination models trained on Multi30k data and the unconstrained data (Section 5.3)
- Transformer text-only models trained on Multi30k (Section 5.2) and the unconstrained data (Section 5.3)
- Transformer multimodal models with serial, parallel, flat and hierarchical input combination strategy trained on Multi30k (Section 5.2)
- Transformer Imagination model trained on Multi30k and the unconstrained data (Section 5.3)

Language Models. We trained an RNN LM with a single Gated Recurrent Unit (GRU) layer (Cho et al., 2014a) of 1,000 dimensions end embeddings of 300. The Transformer LM (Vaswani et al., 2017) has model dimension 512, 6 layers, 8 attention heads and hidden layer size 4,096. The LMs thus mimic the hyperparameters of decoders used in the MMT experiments. We evaluate the models using perplexity on the test part of Multi30k.

For completeness, we also compare the LMs with ELMo (Peters et al., 2018), a representation based on deep RNN LM with character-based embeddings trained on a large corpus of 30 million sentences, and BERT (Devlin et al., 2018), a Transformer-based sentence representation that is similar to Transformer LM. We note however that BERT is trained in a significantly different procedure than regular LMs. We do not train ELMo and BERT ourselves, but rather use the pre-trained models provided by the authors of the respective papers. We neither attempt to compute perplexity of ELMo because it uses a different output vocabulary, nor perplexity of BERT where the architecture does not easily allow such estimate.

BLEU vs. …	Trans.	RNN
image retrieval R@10	.825	.700
STS performance	.852	.873
training data size	.867	.724

Table 6.3: Pearson correlation of MMT quality and representation properties.

Imaginet. The Imaginet models (Chrupała et al., 2015) predict image representation given a sentence and thus train the representations only via its grounding in the image representation.

We use a bidirectional RNN encoder with 300 hidden units in each direction and word embeddings of 300 dimensions. The Transformer-based Imaginet uses the same hyperparameters as the Transformer-based LM. Note that the hyperparameters are the same as in the decoder part of the respective MMT models. The states of the encoder are then mean-pooled and projected with a hidden layer of 4,096 and Rectified Linear Unit (ReLU) non-linearity to a 2,048-dimensional vector corresponding to the image representation from the ResNet (He et al., 2016), in the same way as in the Imagination models.

We evaluate the models on image retrieval on the test part of Multi30k and report Recall at 10. For a fair comparison, we use the representation before the final non-linear projection in the probing experiment and omit the layer that was explicitly trained to fit the image representations.

6.2.4 Results & Discussion

The quantitative results of the image retrieval and STS along with the task-specific metrics are tabulated in Table 6.4. The results show that on moderate-sized datasets, the target language and visual modality provide a stronger training signal for sentence representations than language modeling. The unconstrained variant of the RNN MMT models obtains a similar performance in the STS as the ELMo and BERT models even though the amount of training data was 25 times smaller than for ELMo.

Although the Transformer models achieve a superior translation quality on the MMT tasks, the results on STS suggest that RNN models obtain semantically richer representations. While the text-only RNN translation models perform better on the image retrieval than the Transformer models, Transformer-based Imagination models that are explicitly trained to predict the image representation outperform their RNN counterparts.

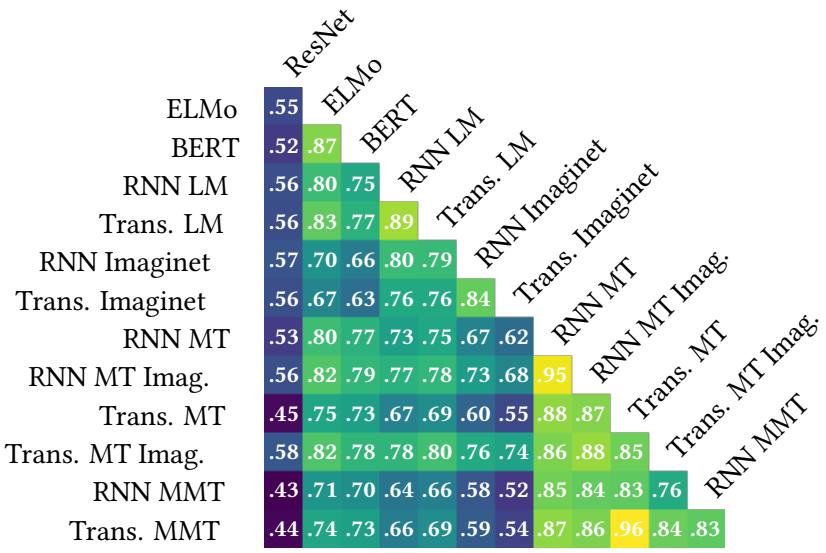


Figure 6.1: Distance correlation of pairs of selected models.

We also evaluate the STS performance of the representations with the CCA projections. The Spearman correlation is consistently worse by about $0.02 - 0.03$. As we notice in Figure 6.1, images have the least DC resulting in information poorer CCA based projections.

The encoder of the MMT models that explicitly use the visual input in the decoder achieves substantially lower image retrieval scores. This observation suggests that the text encoder seems to ignore information about visual aspects of the meaning as the decoder has full access to this information from the explicit conditioning on image representation. This observation is in line with the conclusions of the adversarial evaluation (Elliott, 2018; Libovický et al., 2018a).

Our experiments also indicate that the performance on STS is highly correlated with the translation quality for both the RNN and the Transformer models (see Figure 6.2). However, we observe that the Transformers perform substantially worse with STS than their RNN counterparts. Not surprisingly, the translation quality also appears to be highly correlated not only with the amount of available training data but also with image retrieval abilities of the representation (see Table 6.3).

The result of DC for selected models are shown in Figure 6.1. The DC of the image and the sentence representations is proportional to the image retrieval score. Representations seem to be more similar among the tasks than among the architectures.

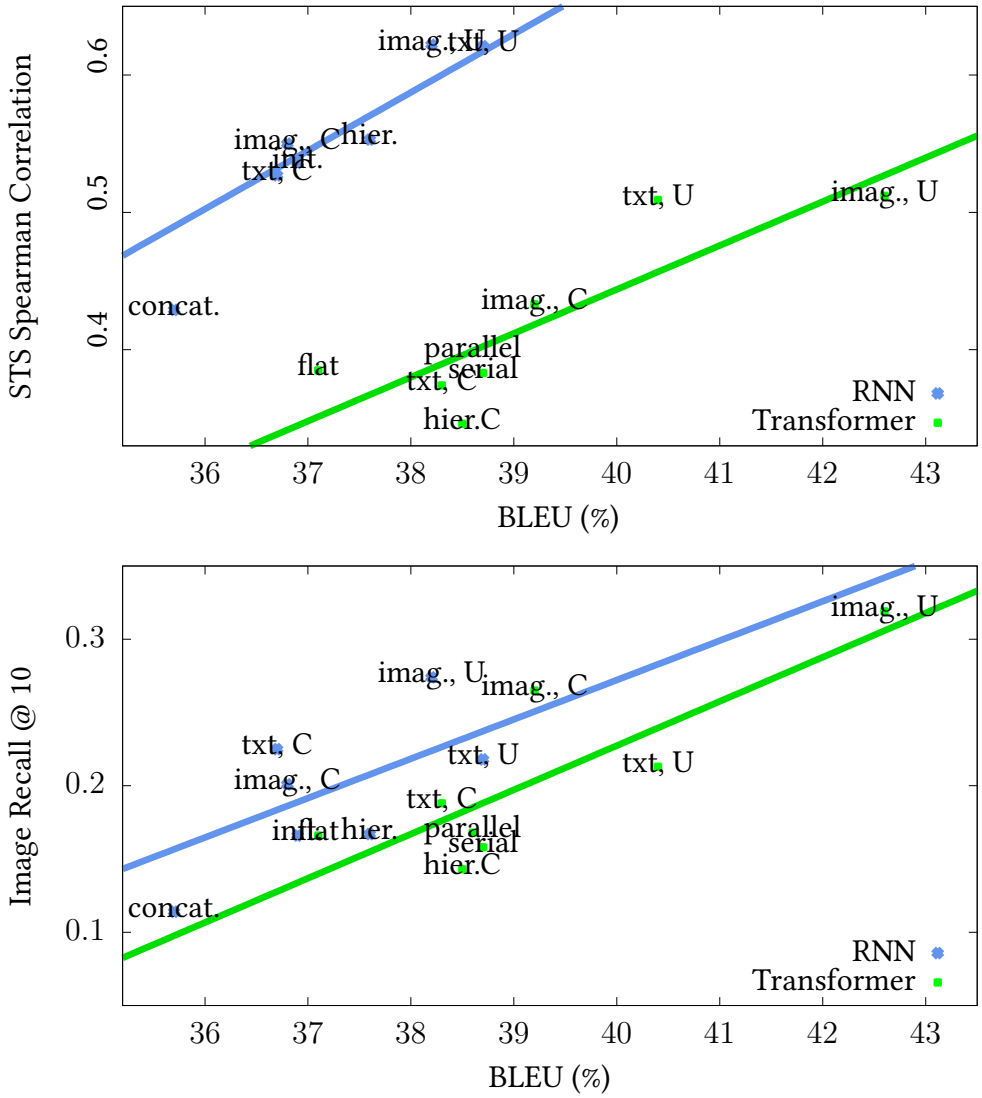


Figure 6.2: Plot of the dependence of BLEU on the Spearman correlation on the STS dataset.

6.2.5 Conclusions

We conducted a set of controlled experiments to asses the representational qualities of monomodal and multimodal sequential models with RNN and SAN architectures. Our experiments show that grounding, in either the visual modality or with another language, and especially their combination in the Imagination models, results in better representations than LMs trained on datasets of similar sizes. We also showed that the translation quality of the MMT models is highly correlated both, with the ability of the models to retain image information and with the semantic properties of the representations. Importantly, we note that the RNN-based models are better at capturing semantics than Transformer-based models under similar training conditions.

Language Model		ppl.↓	Img.↑	STS↑
RNN	Multi30k	12.10	16.6	.267
	Flickr30k	11.80	22.4	.340
	Flickr30k + COCO	11.80	23.0	.378
Transformer	Multi30k	12.42	8.9	.256
	Flickr30k	11.87	17.6	.283
	Flickr30k + COCO	11.69	21.0	.303
ELMo		—	28.4	.631
BERT		—	22.4	.624
Imaginet		R@10↑	Img.↑	STS↑
RNN	Multi30k	29.5	24.4	.401
	Flickr30k	37.8	26.3	.483
	Flickr30k + COCO	39.4	25.4	.501
Transormer	Multi30k	25.5	22.1	.338
	Flickr30k	36.6	29.5	.436
	Flickr30k + COCO	38.4	28.0	.451
Textual MT		BLEU↑	Img.	STS↑
RNN	Textual	36.7	22.5	.527
	Textual U	38.7	21.8	.621
	Imagination	36.8	20.1	.550
	Imagination U	38.2	27.4	.622
Transformer	textual	38.3	18.8	.374
	textual U	40.4	21.3	.509
	Imagination	39.2	26.5	.433
	Imagination U	42.6	31.9	.512
Multimodal MT		BLEU↑	Img.↑	STS↑
RNN	Decoder init.	36.9	16.6	.536
	Att. concatenation	35.7	11.4	.429
	Flat att. comb.	34.6	14.6	.487
	Hierar. att. comb.	37.6	16.7	.553
Transformer	Serial att. comb.	38.7	15.8	.383
	Parallel att. comb.	38.6	16.8	.398
	Flat att. comb.	37.1	16.6	.385
	Hierar. att. comb.	38.5	14.3	.346

Table 6.4: Recall at 10 for image retrieval ('Img.') and Spearman correlation for the Sentence similarity task ('STS') for representation extracted the models. 'U' denotes use of the unconstrained dataset. The first column contains task specific metrics on the Multi30k test set: LM perplexity, image Recall at 10 and BLUE score, respectively.

7
Conclusions

In this thesis, we were concerned with grounding the Natural Language Processing (NLP) models in the visual modality. The popularity of deep learning in Computer Vision (CV) and NLP allowed straightforward reusing of continuous representations between modalities. The models can be trained end-to-end which forces the models to at least to some extent capture how words from language relate to the visual modality. Moreover, they allow solving problems that would otherwise be hardly solvable, such as image captioning or visual question answering. For this thesis, we have chosen the task of Multimodal Machine Translation (MMT). This task is machine translation of image description with the image available as an additional input to the translation model.

We summarized our research on this topic, that included data preparation, introducing novel model architectures and analysis of representation learned by the models. The thesis also summarizes experience from three years of participation in the Workshop of Machine Translation (WMT) shared task on MMT.

Our main contributions are the following:

- We bring a comprehensive overview of deep learning techniques for combing language and vision (Chapter 2).
- We created a Czech version of the MMT benchmark, Multi30k dataset. Additionally, we gathered a large dataset of pseudo-in-domain data that can be used for improving MMT quality (Chapter 4).

- We introduced novel methods for combining multiple sources in sequence-to-sequence models with Recurrent Neural Networks (RNNs) and Self-Attentive Networks (SANs). These methods proved to be useful not only for the MMT task, but they were also adopted by several other authors that used our methods for solving other tasks (Chapter 5).

- We present a thorough analysis of representations that emerge in text-only and multimodal NLP models and show that visual modality provides a strong training signal for semantic properties of the representations (Chapter 6).

The results that we achieved while attempting to solve the task are encouraging and positive even though the multimodal training only brings a modest improvement in translation quality compared to training machine translation on parallel corpora only. The model architectures that we have proposed are able to learn when the visual modality is beneficial for the task and when not. Our analyses also show that the models are at least to some extent capable of capturing some aspects of meaning in the visual modality and that grounding representation in vision improves semantic properties of the representations.

All these findings are encouraging for continuing research on joint modeling of language and other modalities in more challenging and realistic setups which can include tasks like video captions translation (Sanabria et al., 2018) or video summarization (Libovický et al., 2018b).

Bibliography

Agirre, E. – Cer, D. – Diab, M. – Gonzalez-Agirre, A. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, p. 385–393, Montréal, Canada, June 2012. Association for Computational Linguistics.

Agirre, E. – Banea, C. – Cer, D. – Diab, M. – Gonzalez-Agirre, A. – Mihalcea, R. – Rigau, G. – Wiebe, J. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 497–511, San Diego, CA, USA, June 2016. Association for Computational Linguistics.

Amari, S.-i. Backpropagation and stochastic gradient descent method. *Neurocomputing*. Jun 1993, 5, 4-5, p. 185–196. ISSN 0925-2312.

Andrew, G. – Arora, R. – Bilmes, J. – Livescu, K. Deep Canonical Correlation Analysis. In *Proceedings of the 30th International Conference on Machine Learning*, 28 / *Proceedings of Machine Learning Research*, p. 1247–1255, Atlanta, GA, USA, June 2013. PMLR.

Antol, S. – Agrawal, A. – Lu, J. – Mitchell, M. – Batra, D. – Zitnick, L. – Parikh, D. VQA: Visual question answering. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, p. 2425–2433, Arucano Park, Chile, December 2015. IEEE Computer Society.

Ba, L. J. – Kiros, R. – Hinton, G. E. Layer Normalization. *CoRR*. 2016, abs/1607.06450. ISSN 2331-8422.

Bahdanau, D. – Cho, K. – Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*. 2014, abs/1409.0473. ISSN 2331-8422.

Balduzzi, D. – Ghifary, M. Strongly-Typed Recurrent Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, 48 / *Proceedings of Machine Learning Research*, p. 1292–1300, New York, NY, USA, June 2016. PMLR.

Ballard, D. H. – Brown, C. M. *Computer Vision.* Prentice Hall, 1982. ISBN 9780131653160.

Banerjee, S. – Lavie, A. 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*, p. 65–72, Ann Arbor, MI, USA, June 2005. Association for Computational Linguistics.

Barrault, L. – Bougares, F. – Specia, L. – Lala, C. – Elliott, D. – Frank, S. Findings of the Third Shared Task on Multimodal Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, p. 308–327, Brussels, Belgium, October 2018. Association for Computational Linguistics.

Bawden, R. – Sennrich, R. – Birch, A. – Haddow, B. Evaluating Discourse Phenomena in Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, p. 1304–1313, New Orleans, LA, USA, June 2018. Association for Computational Linguistics.

Bengio, Y. – Ducharme, R. – Vincent, P. – Jauvin, C. A neural probabilistic language model. *The Journal of Machine Learning Research*. 2003, 3, Feb, p. 1137–1155. ISSN 1532-4435.

Bengio, Y. – Lamblin, P. – Popovici, D. – Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 20*, p. 153–160, Vancouver, Canada, December 2007. Curran Associates, Inc. ISBN 9780262256919.

Benton, A. – Khayrallah, H. – Gujral, B. – Reisinger, D. – Zhang, S. – Arora, R. Deep Generalized Canonical Correlation Analysis. *CoRR*. 2017, abs/1702.02519. ISSN 2331-8422.

Bojar, O. et al. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, 2, p. 131–198, Berlin, Germany, August 2016a. Association for Computational Linguistics. ISBN 978-1-945626-10-4.

Bojar, O. – Dušek, O. – Kocmi, T. – Libovický, J. – Novák, M. – Popel, M. – Sudarikov, R. – Variš, D. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, no. 9924 in Lecture Notes in Computer Science, p. 231–238, Cham, Switzerland, September 2016b. Springer International Publishing. ISBN 978-3-319-45509-9.

Bojar, O. – Federmann, C. – Fishel, M. – Graham, Y. – Haddow, B. – Koehn, P. – Monz, C. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, p. 272–303, Brussels, Belgium, October 2018. Association for Computational Linguistics.

Branson, S. – Horn, G. V. – Belongie, S. J. – Perona, P. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. *CoRR*. 2014, abs/1406.2952. ISSN 2331-8422.

Caglayan, O. – Aransa, W. – Wang, Y. – Masana, M. – García-Martínez, M. – Bougares, F. – Barrault, L. – Weijer, J. v. d. Does Multimodality Help Human and Machine for Translation and Image Captioning? In *Proceedings of the First Conference on Machine Translation*, p. 627–633, Berlin, Germany, August 2016. Association for Computational Linguistics.

Caglayan, O. – Aransa, W. – Bardet, A. – García-Martínez, M. – Bougares, F. – Barrault, L. – Masana, M. – Herranz, L. – Weijer, J. v. d. LIUM-CVC Submissions for WMT17 Multimodal Translation Task. In *Proceedings of the Second Conference on Machine Translation*, p. 432–439, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Calixto, I. – Liu, Q. Incorporating Global Visual Features into Attention-based Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 992–1003, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Calixto, I. – Liu, Q. – Campbell, N. Doubly-Attentive Decoder for Multi-modal Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1913–1924, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Caruana, R. Multitask Learning. *Machine Learning*. July 1997, 28, 1, p. 41–75. ISSN 0885-6125.

Cavnar, W. B. – Trenkle, J. M. N-Gram-Based Text Categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, p. 161–175, Las Vegas, NV, USA, April 1994. Univeristy of Nevada.

Chan, W. – Jaitly, N. – Le, Q. V. – Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 4960–4964, Shanghai, China, March 2016. IEEE Computer Society. ISBN 9781479999880.

Chen, B. – Cherry, C. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, p. 362–367, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

Chen, D. – Manning, C. D. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 740–750, Doha, Qatar, October 2014. Association for Computational Linguistics.

Chen, M. X. et al. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 76–86, Melbourne, Australia, July 2018. Association for Computational Linguistics.

Chen, X. – Fang, H. – Lin, T.-Y. – Vedantam, R. – Gupta, S. – Dollár, P. – Zitnick, L. Microsoft COCO Captions: Data Collection and Evaluation Server. *CoRR*. 2015, abs/1504.00325. ISSN 2331-8422.

Cho, K. – Merrienboer, B. v. – Bahdanau, D. – Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics.

Cho, K. – Merrienboer, B. v. – Gulcehre, C. – Bahdanau, D. – Bougares, F. – Schwenk, H. – Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics.

Chrupała, G. – Kádár, A. – Alishahi, A. Learning language through pictures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, p. 112–118, Beijing, China, July 2015. Association for Computational Linguistics.

Chrupała, G. – Gelderloos, L. – Alishahi, A. Representations of language in a model of visually grounded speech signal. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 613–622, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Chung, J. – Gülçehre, Ç. – Cho, K. – Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*. 2014, abs/1412.3555. ISSN 2331-8422.

Collobert, R. – Weston, J. – Bottou, L. – Karlen, M. – Kavukcuoglu, K. – Kuksa, P. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*. 2011, 12, Aug, p. 2493–2537. ISSN 1533-7928.

Conneau, A. – Kiela, D. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, p. 1699–1704, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

Conneau, A. – Lample, G. – Ranzato, M. – Denoyer, L. – Jégou, H. Word Translation Without Parallel Data. *CoRR*. 2017, abs/1710.04087. ISSN 2331-8422.

Crevier, D. *AI: the tumultuous history of the search for artificial intelligence*. Basic Books, 1993. ISBN 9780465029976.

Currey, A. – Heafield, K. Multi-Source Syntactic Neural Machine Translation. *CoRR*. 2018, abs/1808.10267. ISSN 2331-8422.

Dabre, R. – Cromierès, F. – Kurohashi, S. Enabling Multi-Source Neural Machine Translation By Concatenating Source Sentences In Multiple Languages. *CoRR*. 2017, abs/1702.06135. ISSN 2331-8422.

Deng, J. – Dong, W. – Socher, R. – Li, L.-J. – Li, K. – Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 248–255, Miami, FL, USA, June 2009. IEEE Computer Society. ISBN 9781424439928.

Denkowski, M. – Lavie, A. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, p. 85–91, Edinburgh, United Kingdom, July 2011. Association for Computational Linguistics.

Deselaers, T. – Ferrari, V. Visual and semantic similarity in imagenet. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1777–1784, Colorado Sprtings, CO, USA, June 2011. IEEE Computer Society. ISBN 978-1-4577-0394-2.

Devlin, J. – Chang, M.-W. – Lee, K. – Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*. 2018, abs/1810.04805. ISSN 2331-8422.

Do, T. – Artieres, T. Neural conditional random fields. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9 / *Proceedings of Machine Learning Research*, p. 177–184, Sardinia, Italy, May 2010. PMLR.

Dorr, B. J. – Jordan, P. W. – Benoit, J. W. A Survey of Current Paradigms in Machine Translation. Technical Report LAMP-TR-027, UMIACS-TR-98-72, CS-TR-3961, University of Maryland, College Park, College Park, MD, USA, December 1998.

Elliott, D. Adversarial Evaluation of Multimodal Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2974–2978, Brussels, Belgium, October 2018. Association for Computational Linguistics.

Elliott, D. – Kádár, A. Imagination Improves Multimodal Translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 130–141, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

Elliott, D. – Kleppe, M. 1 Million Captioned Dutch Newspaper Images. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, p. 3054–3058, Paris, France, May 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Elliott, D. – Frank, S. – Hasler, E. Multi-Language Image Description with Neural Sequence Models. *CoRR*. 2015, abs/1510.04709. ISSN 2331-8422.

Elliott, D. – Frank, S. – Sima'an, K. – Specia, L. Multi30K: Multilingual English-German Image Descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, p. 70–74, Berlin, Germany, August 2016. Association for Computational Linguistics.

Elliott, D. – Frank, S. – Barrault, L. – Bougares, F. – Specia, L. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description. In *Proceedings of the Second Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, p. 215–233, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Elman, J. L. Finding Structure in Time. *Cognitive Science*. 1990, 14, 2, p. 179–211. ISSN 1551-6709.

Erhan, D. – Bengio, Y. – Courville, A. – Vincent, P. Visualizing Higher-Layer Features of a Deep Network. Technical report, Univeristé de Montréal, Montreal, Canada, January 2009.

Faaß, G. – Eckart, K. SdeWaC–a corpus of parsable sentences from the web. In *Language processing and knowledge in the Web*, p. 61–68, Darmstatd, Germany, September 2013. Springer. ISBN 9783642407222.

Fahlman, S. E. – Hinton, G. E. Connectionist Architectures for Artificial Intelligence. *IEEE Computer*. Jan 1987, 20, 1, p. 100–109. ISSN 0018-9162.

Farhadi, A. – Hejrati, M. – Sadeghi, M. A. – Young, P. – Rashtchian, C. – Hockenmaier, J. – Forsyth, D. Every Picture Tells a Story: Generating Sentences from Images. In *Computer Vision – ECCV 2010*, p. 15–29, Berlin, Germany, September 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15561-1.

Firat, O. – Cho, K. Conditional Gated Recurrent Unit with Attention Mechanism. https://github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf, May 2016. Published online, version `adbaeea`.

Firat, O. – Cho, K. – Bengio, Y. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 866–875, San Diego, CA, USA, June 2016. Association for Computational Linguistics.

Frank, S. – Elliott, D. – Specia, L. Assessing multilingual multimodal image description: Studies of native speaker preferences and translator choices. *Natural Language Engineering*. Apr 2018, 24, 3, p. 393–413. ISSN 1469-8110.

Gehring, J. – Auli, M. – Grangier, D. – Yarats, D. – Dauphin, Y. N. Convolutional Sequence to Sequence Learning. In *International Conference on Machine Learning*, p. 1243–1252, Sydney, Australia, August 2017. PMLR.

Gella, S. – Lapata, M. – Keller, F. Unsupervised Visual Sense Disambiguation for Verbs using Multimodal Embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 182–192, San Diego, CA, USA, June 2016. Association for Computational Linguistics.

Gerz, D. – Vulić, I. – Hill, F. – Reichart, R. – Korhonen, A. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2173–2182, Austin, TX, USA, November 2016. Association for Computational Linguistics.

Girshick, R. B. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, p. 1440–1448, Arucano Park, Chile, December 2015. IEEE Computer Society. ISBN 9781467383912.

Gong, Y. – Wang, L. – Hodosh, M. – Hockenmaier, J. – Lazebnik, S. Improving Image-Sentence Embeddings Using Large Weakly Annotated Photo Collections. In *Computer Vision – ECCV 2014*, p. 529–545, Cham, Switzerland, September 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

Goodfellow, I. – Bengio, Y. – Courville, A. *Deep learning*. MIT press, 2016. ISBN 9780262035613.

Goyal, Y. – Khot, T. – Summers-Stay, D. – Batra, D. – Parikh, D. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 6904–6913, Honolulu, HI, USA, July 2017. IEEE Computer Society.

Graves, A. – Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*. Jul 2005, 18, 5-6, p. 602–610. ISSN 0893-6080.

Graves, A. – Schmidhuber, J. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 21*, p. 545–552, Vancouver, Canada, December 2009. Curran Associates, Inc.

Graves, A. – Fernández, S. – Gomez, F. – Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, p. 369–376, Pittsburgh, PA, USA, June 2006. JMLR.org.

Graves, A. – Mohamed, A. – Hinton, G. E. Speech recognition with deep recurrent neural networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 6645–6649, Vancouver, Canada, May 2013. IEEE Computer Society. ISBN 9781479903566.

Graves, A. – Wayne, G. – Danihelka, I. Neural Turing Machines. *CoRR*. 2014, abs/1410.5401. ISSN 2331-8422.

Grönroos, S.-A. – Huet, B. – Kurimo, M. – Laaksonen, J. – Merialdo, B. – Pham, P. – Sjöberg, M. – Sulubacak, U. – Tiedemann, J. – Troncy, R. – Vázquez, R. The MeMAD Submission to the WMT18 Multimodal Translation Task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, p. 609–617, Brussels, Belgium, October 2018. Association for Computational Linguistics.

Hahnloser, R. H. – Sarpeshkar, R. – Mahowald, M. A. – .Douglas, R. J. – Seung, S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*. 2000, 405, 6789, p. 947. ISSN 1476-4687.

Hardoon, D. R. – Szedmak, S. – Shawe-Taylor, J. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*. 2004, 16, 12, p. 2639–2664. ISSN 0899-7667.

Harris, Z. S. Distributional structure. *Word*. 1954, 10, 2-3, p. 146–162. ISSN 0043-7956.

Harwath, D. – Glass, J. Learning Word-Like Units from Joint Audio-Visual Analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 506–517, Vancouver, Canada, July 2017. Association for Computational Linguistics.

He, K. – Zhang, X. – Ren, S. – Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 770–778, Las Vegas, NV, USA, June 2016. IEEE Computer Society. ISBN 9781467388511.

Helcl, J. – Libovický, J. CUNI System for the WMT17 Multimodal Translation Task. In *Proceedings of the Second Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, p. 450–457, Copenhagen, Denmark, September 2017a. Association for Computational Linguistics.

Helcl, J. – Libovický, J. Neural Monkey: An Open-source Tool for Sequence Learning. *The Prague Bulletin of Mathematical Linguistics*. Apr 2017b, 107, 1, p. 5–17. ISSN 0032-6585.

Helcl, J. – Libovický, J. – Kocmi, T. – Musil, T. – Cífka, O. – Variš, D. – Bojar, O. Neural Monkey: The Current State and Beyond. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, p. 168–176, Boston, MA, USA, March 2018a. The Association for Machine Translation in the Americas.

Helcl, J. – Libovický, J. – Variš, D. CUNI System for the WMT18 Multimodal Translation Task. In *Proceedings of the Third Conference on Machine Translation*, p. 622–629, Brussels, Belgium, October 2018b. Association for Computational Linguistics.

Hill, F. – Reichart, R. – Korhonen, A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*. Dec 2015, 41, 4, p. 665–695. ISSN 0891-2017.

Hinton, G. E. – Salakhutdinov, R. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006, 313, 5786, p. 504–507. ISSN 0036-8075.

Hochreiter, S. – Schmidhuber, J. Long short-term memory. *Neural Computation*. 1997, 9, 8, p. 1735–1780. ISSN 0899-7667.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*. 1991, 4, 2, p. 251–257. ISSN 0893-6080.

Hotelling, H. Relations between two sets of variates. *Biometrika*. 1936, 28, 3/4, p. 321–377. ISSN 1464-3510.

Hovy, E. – King, M. – Popescu-Belis, A. An introduction to MT evaluation. In *Proceedings of Machine Translation Evaluation: Human Evaluators meet Automated Metrics. Workshop at the LREC 2002 Conference.*, p. 1–7, Las Palmas, Spain, May 2002. European Language Resources Association.

Hu, J. – Shen, L. – Sun, G. Squeeze-and-Excitation Networks. *CoRR*. 2017, abs/1709.01507. ISSN 2331-8422.

Huang, P.-Y. – Liu, F. – Shiang, S.-R. – Oh, J. – Dyer, C. Attention-based Multimodal Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation*, p. 639–645, Berlin, Germany, August 2016. Association for Computational Linguistics.

Huh, M. – Agrawal, P. – Efros, A. A. What makes ImageNet good for transfer learning? *CoRR*. 2016, abs/1608.08614. ISSN 2331-8422.

Ioffe, S. – Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, p. 448–456, Lille, France, July 2015. JMLR.org.

Jaderberg, M. – Simonyan, K. – Vedaldi, A. – Zisserman, A. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *CoRR*. 2014, abs/1406.2227. ISSN 2331-8422.

Jurafsky, D. – Martin, J. H. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., 2009. ISBN 0131873210.

Kalchbrenner, N. – Blunsom, P. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1700–1709, Seattle, WA, USA, October 2013. Association for Computational Linguistics.

Kessler, B. – Nunberg, G. – Schütze, H. Automatic Detection of Text Genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, p. 32–38, Madrid, Spain, July 1997. Association for Computational Linguistics.

Keysers, D. – Deselaers, T. – Rowley, H. A. – Wang, L.-L. – Carbune, V. Multi-Language Online Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* June 2017, 39, 6, p. 1180–1194. ISSN 0162-8828.

Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

Kingma, D. P. – Ba, J. Adam: A Method for Stochastic Optimization. *CoRR*. 2014, abs/1412.6980. ISSN 2331-8422.

Kingma, D. P. – Mohamed, S. – Rezende, D. J. – Welling, M. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27*, p. 3581–3589, Montreal, Canada, December 2014. Curran Associates, Inc.

Kiperwasser, E. – Goldberg, Y. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *Transactions of the Association for Computational Linguistics*. Dec 2016, 4, p. 313–327. ISSN 2307-387X.

Klein, G. – Kim, Y. – Deng, Y. – Nguyen, V. – Senellart, J. – Rush, A. OpenNMT: Neural Machine Translation Toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, p. 177–184, Boston, MA, USA, March 2018. Association for Machine Translation in the Americas.

Koehn, P. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Koehn, P. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*, 5, p. 79–86, Phuket, Thailand, September 2005. Asia-Pacific Association for Machine Translation.

Koehn, P. – Knowles, R. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, p. 28–39, Vancouver, Canada, August 2017. Association for Computational Linguistics.

Koehn, P. – Hoang, H. – Birch, A. – Callison-Burch, C. – Federico, M. – Bertoldi, N. – Cowan, B. – Shen, W. – Moran, C. – Zens, R. – Dyer, C. – Bojar, O. – Constantin, A. – Herbst, E. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, p. 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Krizhevsky, A. – Sutskever, I. – Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 60, p. 1097–1105, Red Hook, NY, USA, May 2012. Curran Associates, Inc. ISBN 9781627480031.

Kulkarni, G. – Premraj, V. – Dhar, S. – Li, S. – Choi, Y. – Berg, A. C. – Berg, T. L. Baby talk: Understanding and generating simple image descriptions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1601–1608, Colorado Sprtings, CO, USA, June 2011. IEEE Computer Society. ISBN 978-1-4577-0394-2.

Lafferty, J. – McCallum, A. – Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*, p. 282–289, Williamstown, MA, USA, June 2001. Morgan Kaufmann. ISBN 1-55860-778-1.

Lample, G. – Ballesteros, M. – Subramanian, S. – Kawakami, K. – Dyer, C. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*, p. 260–270, San Diego, CA, USA, June 2016. Association for Computational Linguistics.

Lawrance, A. – Lewis, P. An exponential moving-average sequence and point process (EMA1). *Journal of Applied Probability*. 1977, 14, 1, p. 98–113. ISSN 0021-9002.

Lazaridou, A. – Pham, N. T. – Baroni, M. Combining Language and Vision with a Multimodal Skip-gram Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 153–163, Denver, CO, USA, June 2015. Association for Computational Linguistics.

Le, Q. V. – Schuster, M. A Neural Network for Machine Translation, at Production Scale, 2016.

LeCun, Y. – Bottou, L. – Bengio, Y. – Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, 86, 11, p. 2278–2324. ISSN 0018-9219.

LeCun, Y. – Bengio, Y. – Hinton, G. E. Deep learning. *Nature*. 2015, 521, 7553, p. 436–444. ISSN 1476-4687.

Lee, J. – Cho, K. – Hofmann, T. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics*. Dec 2017a, 5, p. 365–378. ISSN 2307-387X.

Lee, K. – Levy, O. – Zettlemoyer, L. Recurrent Additive Networks. *CoRR*. 2017b, abs/1705.07393. ISSN 2331-8422.

Lee, Y.-B. – Myaeng, S. H. Text Genre Classification with Genre-revealing and Subject-revealing Features. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, p. 145–150, New York, NY, USA, August 2002. ACM. ISBN 1-58113-561-0.

Levy, O. – Goldberg, Y. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, p. 2177–2185, Montreal, Canada, December 2014. Curran Associates, Inc.

Li, H. – Zhu, J. – Liu, T. – Zhang, J. – Zong, C. Multi-modal Sentence Summarization with Modality Attention and Image Filtering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, p. 4152–4158, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 9780999241127.

Li, S. – Kulkarni, G. – Berg, T. L. – Berg, A. C. – Choi, Y. Composing Simple Image Descriptions using Web-scale N-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, p. 220–228, Portland, OR, USA, June 2011. Association for Computational Linguistics.

Libovický, J. – Helcl, J. Attention Strategies for Multi-Source Sequence-to-Sequence Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 196–202, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Libovický, J. – Helcl, J. – Tlustý, M. – Bojar, O. – Pecina, P. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, p. 646–654, Berlin, Germany, August 2016. Association for Computational Linguistics.

Libovický, J. – Helcl, J. – Mareček, D. Input Combination Strategies for Multi-Source Transformer Decoder. In *Proceedings of the Third Conference on Machine Translation*, p. 253–260, Brussels, Belgium, October 2018a. Association for Computational Linguistics.

Libovický, J. – Palaskar, S. – Gella, S. – Metze, F. Multimodal Abstractive Summarization for Open-Domain Videos. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*, Montreal, Canada, December 2018b. Neural Information Processing Systems Foundation.

Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, p. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Lin, T.-Y. – Maire, M. – Belongie, S. J. – Hays, J. – Perona, P. – Ramanan, D. – Dollár, P. – Zitnick, L. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, p. 740–755, Cham, Switzerland, September 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Lin, Z. – Feng, M. – Santos, C. N. d. – Yu, M. – Xiang, B. – Zhou, B. – Bengio, Y. A Structured Self-attentive Sentence Embedding. *CoRR*. 2017, abs/1703.03130. ISSN 2331-8422.

Ling, W. – Dyer, C. – Black, A. W. – Trancoso, I. – Fermandez, R. – Amir, S. – Marujo, L. – Luis, T. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Lopez, A. Statistical Machine Translation. *ACM Computing Surveys*. September 2008, 40, 3, p. 8:1–8:49. ISSN 0360-0300.

Lu, J. – Yang, J. – Batra, D. – Parikh, D. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *Advances in Neural Information Processing Systems 29*, p. 289–297, Barcelona, Spain, December 2016. Curran Associates, Inc.

Lu, J. – Xiong, C. – Parikh, D. – Socher, R. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 375–383, Honolulu, HI, USA, July 2017. IEEE Computer Society. ISBN 9781538604571.

Luong, T. – Pham, H. – Manning, C. D. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, p. 151–159, Denver, CO, USA, June 2015a. Association for Computational Linguistics.

Luong, T. – Pham, H. – Manning, C. D. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1412–1421, Lisbon, Portugal, September 2015b. Association for Computational Linguistics.

Madhyastha, P. – Wang, J. – Specia, L. Sheffield MultiMT: Using Object Posterior Predictions for Multimodal Machine Translation. In *Proceedings of the Second Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, p. 470–476, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Madhyastha, P. – Wang, J. – Specia, L. End-to-end image captioning exploits multimodal distributional similarity. In *29th British Machine Vision Conference*, p. 1–13, Newcastle, United Kingdom, September 2018. British Machine Vision Association.

Mahendran, A. – Vedaldi, A. Understanding deep image representations by inverting them. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 233–255, Boston, MA, USA, June 2015. IEEE Computer Society. ISBN 9781467369640.

Manning, C. D. – Schütze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. ISBN 0-262-13360-1.

Marmanis, D. – Datcu, M. – Esch, T. – Stilla, U. Deep learning earth observation classification using ImageNet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*. Jan 2016, 13, 1, p. 105–109. ISSN 1545598X.

McCulloch, W. S. – Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*. Dec 1943, 5, 4, p. 115–133. ISSN 1522-9602.

Mikolov, T. – Karafiát, M. – Burget, L. – Černockỳ, J. – Khudanpur, S. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, p. 1045–1048, Makuhari, Japan, September 2010. International Speech Communication Association.

Mikolov, T. – Yih, W.-t. – Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 746–751, Atlanta, GA, USA, June 2013. Association for Computational Linguistics.

Miller, G. A. WordNet: a lexical database for English. *Communications of the ACM*. 1995, 38, 11, p. 39–41. ISSN 0001-0782.

Miltenburg, E. v. Stereotyping and Bias in the Flickr30K Dataset. *CoRR*. 2016, abs/1605.06083. ISSN 2331-8422.

Mitchell, M. – Dodge, J. – Goyal, A. – Yamaguchi, K. – Stratos, K. – Han, X. – Mensch, A. – Berg, A. – Berg, T. L. – III, H. D. Midge: Generating Image Descriptions From Computer Vision Detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 747–756, Avignon, France, April 2012. Association for Computational Linguistics.

Nair, V. – Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, p. 807–814, Haifa, Israel, June 2010. JMLR.org.

Neves, M. – Yepes, A. J. – Névéol, A. – Grozea, C. – Siu, A. – Kittner, M. – Verspoor, K. Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on Medline test sets. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, p. 328–343, Belgium, Brussels, October 2018. Association for Computational Linguistics.

Nishimura, Y. – Sudoh, K. – Neubig, G. – Nakamura, S. Multi-Source Neural Machine Translation with Missing Data. In *The Second Workshop on Neural Machine Translation and Generation (WNMT)*, p. 92–99, Melbourne, Australia, July 2018. Association for Computational Linguistics.

Novikoff, A. B. On Convergence Proofs on Perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, 12, p. 615–622, New York, NY, USA, April 1962. Polytechnic Institute of Brooklyn.

Olah, C. – Mordvintsev, A. – Schubert, L. Feature Visualization. *Distill*. Nov 2017, 2, 11, p. e7. ISSN 2476-0757. https://distill.pub/2017/feature-visualization.

O'Regan, G. *Introduction to the History of Computing: A Computing History Primer*. Undergraduate Topics in Computer Science. Springer International Publishing, 2016. ISBN 9783319331386.

Pak, A. – Paroubek, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, 5, p. 1320–1326, Valletta, Malta, May 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Pang, B. – Lee, L. – Vaithyanathan, S. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 79–86, Philadelphia, PA, USA, July 2002. Association for Computational Linguistics.

Papineni, K. – Roukos, S. – Ward, T. – Zhu, W.-J. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, PA, USA, July 2002. Association for Computational Linguistics.

Parikh, A. – Täckström, O. – Das, D. – Uszkoreit, J. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2249–2255, Austin, TX, USA, November 2016. Association for Computational Linguistics.

Parkhi, O. M. – Vedaldi, A. – Zisserman, A. Deep Face Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, p. 41.1–41.12, Swansea, United Kingdom, September 2015. British Machine Vision Association. ISBN 1-901725-53-7.

Pascanu, R. – Mikolov, T. – Bengio, Y. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, p. 1310–1318, Atlanta, GA, USA, June 2013. PMLR.

Peters, M. – Neumann, M. – Iyyer, M. – Gardner, M. – Clark, C. – Lee, K. – Zettlemoyer, L. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, p. 2227–2237, New Orleans, LA, USA, June 2018. Association for Computational Linguistics.

Plummer, B. A. – Wang, L. – Cervantes, C. M. – Caicedo, J. C. – Hockenmaier, J. – Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, p. 2641–2649, Les Condres, Chile, December 2015. IEEE Computer Society.

Poerner, N. – Roth, B. – Schütze, H. Interpretable Textual Neuron Representations for NLP. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 325–327, Brussels, Belgium, November 2018. Association for Computational Linguistics.

Pradhan, S. S. – Hovy, E. – Marcus, M. – Palmer, M. – Ramshaw, L. – Weischedel, R. OntoNotes: A Unified Relational Semantic Representation. In *Proceedings of the International Conference on Semantic Computing*, p. 517–526, Irvine, CA, USA, September 2007. IEEE Computer Society.

Press, O. – Wolf, L. Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics.

Qi, Y. – Sachan, D. – Felix, M. – Padmanabhan, S. – Neubig, G. When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, p. 529–535, New Orleans, LA, USA, June 2018. Association for Computational Linguistics.

Rashtchian, C. – Young, P. – Hodosh, M. – Hockenmaier, J. 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*, p. 139–147, Los Angeles, CA, USA, June 2010. Association for Computational Linguistics.

Redmon, J. – Divvala, S. – Girshick, R. B. – Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 779–788, Las Vegas, NV, USA, June 2016. IEEE Computer Society. ISBN 9781467388511.

Reed, S. E. – Akata, Z. – Yan, X. – Logeswaran, L. – Schiele, B. – Lee, H. Generative Adversarial Text-to-Image Synthesis. In *Proceedings of the 33rd International Conference on Machine Learning*, 48, p. 1060–1069, New York, NY, USA, June 2016. PMLR.

Ren, S. – He, K. – Girshick, R. B. – Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*, 39, p. 91–99, Montreal, Canada, December 2015. Curran Associates, Inc.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*. 1958, 65, 6, p. 386–408. ISSN 0033-295X.

Sanabria, R. – Caglayan, O. – Palaskar, S. – Elliott, D. – Barrault, L. – Specia, L. – Metze, F. How2: A Large-scale Dataset for Multimodal Language Understanding. *CoRR*. 2018, abs/1811.00347. ISSN 2331-8422.

Sánchez, J. – Perronnin, F. High-dimensional signature compression for large-scale image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1665–1672, Colorado Springs, CO, USA, June 2011. IEEE Computer Society. ISBN 9781457703942.

Saphra, N. – Lopez, A. Language Models Learn POS First. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 328–330, Brussels, Belgium, November 2018. Association for Computational Linguistics.

Saussure, F. d. *Course in General Linguistics*. Philosophical Library, 1916. ISBN 9781330033586.

Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *CoRR*. 2014, abs/1404.7828. ISSN 2331-8422.

Schroff, F. – Kalenichenko, D. – Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 815–823, Boston, MA, USA, June 2015. IEEE Computer Society. ISBN 9781467369640.

Schuster, M. – Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*. 1997, 45, 11, p. 2673–2681. ISSN 1053-587X.

Sennrich, R. – Haddow, B. – Birch, A. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 86–96, Berlin, Germany, August 2016a. Association for Computational Linguistics.

Sennrich, R. – Haddow, B. – Birch, A. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1715–1725, Berlin, Germany, August 2016b. Association for Computational Linguistics.

Sennrich, R. – Firat, O. – Cho, K. – Birch, A. – Haddow, B. – Hitschler, J. – Junczys-Dowmunt, M. – Läubli, S. – Barone, A. V. M. – Mokry, J. – Nadejde, M. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, p. 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics.

Seo, M. J. – Kembhavi, A. – Farhadi, A. – Hajishirzi, H. Bidirectional Attention Flow for Machine Comprehension. *CoRR*. 2016, abs/1611.01603. ISSN 2331-8422.

Shah, K. – Wang, J. – Specia, L. SHEF-Multimodal: Grounding Machine Translation on Images. In *Proceedings of the First Conference on Machine Translation*, p. 660–665, Berlin, Germany, August 2016. Association for Computational Linguistics.

Shaw, P. – Uszkoreit, J. – Vaswani, A. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, p. 464–468, New Orleans, LA, USA, June 2018. Association for Computational Linguistics.

Siegelmann, H. T. – Sontag, E. D. On the computational power of neural nets. *Journal of computer and system sciences*. 1995, 50, 1, p. 132–150. ISSN 0022-0000.

Silberer, C. – Lapata, M. Learning Grounded Meaning Representations with Autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 721–732, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

Simonyan, K. – Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*. 2014, abs/1409.1556. ISSN 2331-8422.

Skadiņš, R. – Tiedemann, J. – Rozis, R. – Deksne, D. Billions of Parallel Words for Free: Building and Using the EU Bookshop Corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, p. 1850–1855, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

Smith, J. R. – Saint-Amand, H. – Plamada, M. – Koehn, P. – Callison-Burch, C. – Lopez, A. Dirt Cheap Web-Scale Parallel Text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1374–1383, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Snover, M. – Dorr, B. J. – Schwartz, R. – Micciulla, L. – Makhoul, J. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, p. 223–231, Cambridge, MA, USA, August 2006. The Association for Machine Translation in the Americas.

Sonka, M. – Hlavac, V. – Boyle, R. *Image Processing, Analysis, and Machine Vision.* Thomson-Engineering, 2007. ISBN 049508252X.

Specia, L. – Frank, S. – Sima'an, K. – Elliott, D. A Shared Task on Multimodal Machine Translation and Crosslingual Image Description. In *Proceedings of the First Conference on Machine Translation*, p. 543–553, Berlin, Germany, August 2016. Association for Computational Linguistics.

Srivastava, N. – Hinton, G. E. – Krizhevsky, A. – Sutskever, I. – Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014, 15, 1, p. 1929–1958. ISSN 1532-4435.

Srivastava, R. K. – Greff, K. – Schmidhuber, J. Highway Networks. *CoRR*. 2015, abs/1505.00387. ISSN 2331-8422.

Stewart, D. – Kuhn, R. – Joanis, E. – Foster, G. Coarse split and lump bilingual language models for richer source information in SMT. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*, 1, p. 28–41, Vancouver, Canada, October 2014. The Association for Machine Translation in the Americas.

Straka, M. – Straková, J. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, p. 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics.

Sundermeyer, M. – Schlüter, R. – Ney, H. LSTM neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*, p. 194–197, Portland, OR, USA, September 2012. International Speech Communication Association.

Sutskever, I. – Vinyals, O. – Le, Q. V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, p. 3104–3112, Montreal, Canada, December 2014. Curran Associates, Inc.

Szegedy, C. – Liu, W. – Jia, Y. – Sermanet, P. – Reed, S. E. – Anguelov, D. – Erhan, D. – Vanhoucke, V. – Rabinovich, A. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1–9, Boston, MA, USA, June 2015. IEEE Computer Society. ISBN 9781467369640.

Székely, G. J. – Rizzo, M. L. – Bakirov, N. K. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*. Dec 2007, 35, 6, p. 2769–2794. ISSN 0090-5364.

Tiedemann, J. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, p. 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Toyama, J. – Misono, M. – Suzuki, M. – Nakayama, K. – Matsuo, Y. Neural Machine Translation with Latent Semantic of Image and Text. *CoRR*. 2016, abs/1611.08459. ISSN 2331-8422.

Turchi, M. – Chatterjee, R. – Negri, M. WMT16 APE Shared Task Data, 2016. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

Turing, A. M. Computing machinery and intelligence. *Mind*. 1950, 59, 236, p. 433–460. ISSN 0026-4423.

Vaswani, A. – Shazeer, N. – Parmar, N. – Uszkoreit, J. – Jones, L. – Gomez, A. N. – Kaiser, Ł. – Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, p. 6000–6010, Long Beach, CA, USA, December 2017. Curran Associates, Inc.

Vinyals, O. – Toshev, A. – Bengio, S. – Erhan, D. Show and tell: A neural image caption generator. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 00, p. 3156–3164, Boston, MA, USA, June 2015. IEEE Computer Society. ISBN 9781467369640.

Vinyals, O. – Toshev, A. – Bengio, S. – Erhan, D. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*. Apr 2017, 39, 4, p. 652–663. ISSN 1939-3539.

Wang, A. – Singh, A. – Michael, J. – Hill, F. – Levy, O. – Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 353–355, Brussels, Belgium, November 2018a. Association for Computational Linguistics.

Wang, L. – Li, Y. – Lazebnik, S. Learning Two-Branch Neural Networks for Image-Text Matching Tasks. *CoRR*. 2017, abs/1704.03470. ISSN 2331-8422.

Wang, X. – Li, R. – Hermansky, H. Stream Attention for Distributed Multi-Microphone Speech Recognition. In *Proceedings of Interspeech 2018*, p. 3033–3037, Hyderabad, India, September 2018b. International Speech Communication Association.

Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*. 1990, 78, 10, p. 1550–1560. ISSN 0018-9219.

Widrow, B. Adaptive "Adaline" neuron using chemical "memistors". Technical Report Technical Report 1553-2, Stanford Electron. Labs., Stanford, CA, USA, October 1960.

Wu, F. – Lao, N. – Blitzer, J. – Yang, G. – Weinberger, K. Q. Fast Reading Comprehension with ConvNets. *CoRR*. 2017, abs/1711.04352. ISSN 2331-8422.

Wu, Y. et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*. 2016, abs/1609.08144. ISSN 2331-8422.

Xie, S. – Girshick, R. B. – Dollár, P. – Tu, Z. – He, K. Aggregated Residual Transformations for Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 5987–5995, Honolulu, HI, USA, July 2017. IEEE Computer Society. ISBN 9781538604571.

Xu, K. – Ba, J. – Kiros, R. – Cho, K. – Courville, A. – Salakhutdinov, R. – Zemel, R. – Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning*, 37 / *Proceedings of Machine Learning Research*, p. 2048–2057, Lille, France, July 2015. PMLR.

Yan, F. – Mikolajczyk, K. Deep correlation for matching images and text. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 3441–3450, Boston, MA, USA, June 2015. IEEE Computer Society. ISBN 9781467369640.

Yasuda, K. – Zhang, R. – Yamamoto, H. – Sumita, E. Method of selecting training data to build a compact and efficient translation model. In *Proceedings of the Third International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, p. 655–660, Hyderabad, India, January 2008. Asian Federation of Natural Language Processing.

Yoshino, K. – Hori, C. – Perez, J. – D'Haro, L. F. – Polymenakos, L. – Gunasekara, C. – Lasecki, W. S. – Kummerfeld, J. K. – Galley, M. – Brockett, C. – others. Dialog System Technology Challenge 7. In *Proceeings of the 2nd Conversational AI Workshop*, Montreal, Canada, December 2018. Neural Information Processing Systems Foundation.

Young, P. – Lai, A. – Hodosh, M. – Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*. Dec 2014, 2, p. 67–78. ISSN 2307-387X.

Yu, F. – Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *CoRR*. 2015, abs/1511.07122. ISSN 2331-8422.

Zeiler, M. D. – Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, p. 818–833, Zurich, Switzerland, September 2014. Springer. ISBN 9783319105901.

Zhang, P. – Goyal, Y. – Summers-Stay, D. – Batra, D. – Parikh, D. Yin and Yang: Balancing and Answering Binary Visual Questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 5014–5022, Las Vegas, NV, USA, June 2016. IEEE Computer Society. ISBN 9781467388511.

Zhu, X. – Li, L. – Liu, J. – Peng, H. – Niu, X. Captioning Transformer with Stacked Attention Modules. *Applied Sciences*. May 2018, 8, 5, p. 1–11. ISSN 2076-3417.

Zhu, Z. – Liang, D. – Zhang, S. – Huang, X. – Li, B. – Hu, S. Traffic-Sign Detection and Classification in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 2110–2118, Las Vegas, NV, USA, June 2016. IEEE Computer Society. ISBN 9781467388511.

Zoph, B. – Knight, K. Multi-Source Neural Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 30–34, San Diego, CA, USA, June 2016. Association for Computational Linguistics.

List of Publications

Libovický, J. – Helcl, J. – Tlustý, M. – Bojar, O. – Pecina, P. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, p. 646–654, Berlin, Germany, August 2016. Association for Computational Linguistics

- The paper describes a submission to the WMT16 which describes our early experiments with Multimodal Machine Translation (MMT). This paper is partially discussed in Sections 4.3 and 5.1.
- Citations (without self-citations): 28

Libovický, J. – Helcl, J. Attention Strategies for Multi-Source Sequence-to-Sequence Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 196–202, Vancouver, Canada, July 2017. Association for Computational Linguistics

- The paper introduces techniques for combining multiple different inputs in sequence-to-sequence learning using Recurrent Neural Networks (RNNs). Content of this paper is discussed mainly in Section 5.1.
- Awarded as an Outstanding Paper at ACL 2017 and ÚFAL Best Paper 2017.
- Citations (without self-citations): 19

Helcl, J. – Libovický, J. Neural Monkey: An Open-source Tool for Sequence Learning. *The Prague Bulletin of Mathematical Linguistics*. Apr 2017b, 107, 1, p. 5–17. ISSN 0032-6585

- This paper introduces a software tool *Neural Monkey* which was used for all the experiments in this thesis.
- Citations (without self-citations): 22

Helcl, J. – Libovický, J. CUNI System for the WMT17 Multimodal Translation Task. In *Proceedings of the Second Conference on Machine Translation (WMT), Volume 2: Shared Task Papers*, p. 450–457, Copenhagen, Denmark, September 2017a. Association for Computational Linguistics

- A submission to WMT17 MMT task. The submission tests architectures proposed in the previous paper in a more competitive setup and discusses techniques for acquiring additional training data which are discussed in Section 5.3.
- Citations (without self-citations): 7

Helcl, J. – Libovický, J. – Kocmi, T. – Musil, T. – Cífka, O. – Variš, D. – Bojar, O. Neural Monkey: The Current State and Beyond. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, p. 168–176, Boston, MA, USA, March 2018a. The Association for Machine Translation in the Americas

- A paper summarizing the Neural Monkey software at the beginning of 2018.
- Citations (without self-citations): 1

Libovický, J. – Helcl, J. – Mareček, D. Input Combination Strategies for Multi-Source Transformer Decoder. In *Proceedings of the Third Conference on Machine Translation*, p. 253–260, Brussels, Belgium, October 2018a. Association for Computational Linguistics

- The paper introduces techniques for input combinations in sequence-to-sequence models with self-attentive encoder and decoder.
- Citations (without self-citations): 0

Helcl, J. – Libovický, J. – Variš, D. CUNI System for the WMT18 Multimodal Translation Task. In *Proceedings of the Third Conference on Machine Translation*, p. 622–629, Brussels, Belgium, October 2018b. Association for Computational Linguistics

- A paper summarizing the Neural Monkey software at the beginning of 2018.
- Citations (without self-citations): 0

Only publication relevant to this thesis are included. The number of citations was computed using Google Scholar. Total number of citations of publication related to the topic of the thesis (without self-citations): 77

List of Abbreviations

AI Artificial Intelligence. 1, 7
APE Automatic MT Post-Editing. 66, 71, 73–75, 137

BPE Byte-Pair Encoding. 70, 71, 92

CCA Canonical Correlation Analysis. 46, 47, 100–102, 105
CNN Convolutional Neural Network. 11, 14, 18, 21, 25–28, 30, 32, 35, 36, 38, 46–49, 73, 139
CV Computer Vision. 2, 3, 5, 6, 8, 10, 11, 13, 17, 18, 23, 25–27, 43, 45, 47, 48, 109

DC Distance Correlation. 99, 101, 102, 105

GRU Gated Recurrent Unit. 25, 67, 69–71, 75, 88, 92, 103

LM Language Model. 19–21, 35–37, 44, 46, 62, 67, 88, 99, 102–104, 107, 108, 139
LSTM Long Short-Term Memory. 23–26, 36, 49, 67

ML Machine Learning. 5–7, 10, 11
MMT Multimodal Machine Translation. 2, 3, 28, 43, 45, 47, 49, 51, 54–58, 60–66, 69, 72–75, 81–85, 87, 92, 95, 99, 101–105, 107, 109, 110, 133, 134, 137, 140
MT Machine Translation. 10, 21, 28, 35, 36, 38, 45, 47, 48, 51, 52, 54, 55, 61–63, 74–77, 83, 84, 86–88, 99, 140

NLP Natural Language Processing. 1–6, 8, 10, 11, 18–22, 25–27, 33, 35, 43–45, 99, 109, 110

ReLU Rectified Linear Unit. 14, 15, 23, 49, 91, 104
RNN Recurrent Neural Network. 18, 21–26, 28, 33, 35–38, 46–49, 61–65, 67, 69, 72, 76, 77, 79–83, 88, 89, 92, 95, 97, 99, 100, 103–105, 107, 110, 133, 137, 139, 140

SAN Self-Attentive Network. 18, 21, 28, 30, 32, 35, 38, 49, 95, 99, 100, 107, 110
STS Semantic Textual Similarity. 99, 101, 104, 105

WMT Workshop of Machine Translation. 3, 51, 52, 56, 58, 62, 64, 70, 75, 87, 89, 109

List of Tables

List of Figures