THÈSE DE DOCTORAT DE

LE MANS UNIVERSITÉ
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N° 601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

« Ozan CAGLAYAN »

« Multimodal Machine Translation »

Thèse présentée et soutenue à LE MANS UNIVERSITÉ, le 27 Août 2019
Unité de recherche : Laboratoire d'Informatique de l'Université du Mans (LIUM)
Thèse N° : 2019LEMA1016

Rapporteurs avant soutenance :

Alexandre ALLAUZEN	Professeur, LIMSI-CNRS, Université Paris-Sud
Marie-Francine MOENS	Professeur, KU Leuven

Composition du Jury :

Président :	Joost VAN DE WEIJER, PhD, Universitat Autònoma de Barcelona
Examinateur :	Deniz YÜRET, Professeur, Koc University
Dir. de thèse :	Paul DELEGLISE, Professeur Émérite, LIUM, Le Mans Université
Co-dir. de thèse :	Loïc BARRAULT, Maître de Conférences, LIUM, Le Mans Université
Invité(s):	Fethi BOUGARES, Maître de Conférences, LIUM, Le Mans Université

Acknowledgements

Everything began in summer 2014, after completing the online machine learning course of Andrew Ng. I sent an e-mail to Yoshua Bengio, asking tons of questions about doing a PhD in deep learning. He replied by adding Holger Schwenk to the conversation, who was searching for a PhD student in machine translation. So first of all, I am grateful to Andrew Ng, Yoshua Bengio and Holger Schwenk for making this journey possible. Second, I thank all the people from Le Mans University, especially my supervisors Loïc Barrault, Fethi Bougares, Paul Deleglise and my colleagues Walid Aransa, Adrien Bardet, Anne Cécile, Nicolas Dugué, Grégor Dupuy, Yannick Estève, Mercedes García-Martínez, Mélanie Hardy, Malik Koné, and Etienne Micoulaut. I am particularly grateful to Gokcen Eraslan and Orhan Firat for their support, guidance and valuable comments since the early days of this journey. Finally, I would like to thank people from JSALT2018 S2S team for that extraordinary summer in Baltimore.

I would like to thank my colleagues & friends Burak Arslan, Gözde Aytemur, İlke Bereketli, Emre Doğan, Özlem Durmaz İncel, Fatma Gül Karagöz, Burcu Konakçı, Günce Orman, Cem Özatalay, Atay Özgövde, Selin Pelek, Doğu Toksöz, Pınar Uluer and Merve Ünlü from Galatasaray University, for supporting me all the time throughout this long journey which began in Turkey and ended up in France. My deepest appreciation goes to my high school friends who are always around in one way or another: Canan Akyüz, Sibel Aydın, İpek Dağlı, Hande Erguner, Alp Kılıç, Duygu Kuzuoğlu, Ezgi Tunç, Çiğdem Vaizoğlu and Emre Yetkin. I would like to express my gratitude to Alper Açık, Nihan Aksakallı, Tuna Altınel, Aslı Aydemir, Begüm Başdaş, Can Candan, Yonca Demir, Nazım Dikbaş, Hülya Dinçer, Elif Ege, Aslı Odman, Özlem Özkan, Aslı Takanay, Ceren Uysal and numerous other academics for peace. Many thanks to Gülşah Kurt and Ceyda Sungur, for being patient listeners and my anchors to reality in France. I would like to thank Asena Pala for being around during the last weeks of the heavy writing period, which would have otherwise been quite boring. Finally, this thesis would not have been possible without the support and encouragements from my parents.

Thanks Elif :)
"🌸 dıt dıt dıt bu tez çıkış yapıyor, fıtıfıtı da arkasından geliyor hey hey hey! 🌸"

In 2016, more than 2000 academics – including myself – signed a petition for peace, calling the authorities to end the violence in eastern Turkey. Since then, more than 500 signatories have been discharged from their positions, and more than 700 of them have been put on trial for "terrorist propaganda".

As I complete the final words of my thesis, I would like to dedicate my work to two academics for peace: Dr. Mehmet Fatih Traş, who committed suicide after being discharged from his university, and Prof. Füsun Üstel from Galatasaray University who is about to enter prison.

Résumé

La traduction automatique vise à traduire des documents d'une langue à une autre sans l'intervention humaine. Avec l'apparition des réseaux de neurones profonds (DNN), la traduction automatique neuronale (NMT) a commencé à dominer le domaine, atteignant l'état de l'art pour de nombreuses langues. NMT a également ravivé l'intérêt pour la traduction basée sur l'*interlangue* grâce à la manière dont elle place la tâche dans un cadre encodeur-décodeur en passant par des représentations latentes. Combiné avec la flexibilité architecturale des DNN, ce cadre a aussi ouvert une piste de recherche sur la multimodalité, ayant pour but d'enrichir les représentations latentes avec d'autres modalités telles que la vision ou la parole, par exemple. Cette thèse se concentre sur la traduction automatique multimodale (MMT) en intégrant la vision comme une modalité secondaire afin d'obtenir une meilleure compréhension du langage, ancrée de façon visuelle. J'ai travaillé spécifiquement avec un ensemble de données contenant des images et leurs descriptions traduites, où le contexte visuel peut être utile pour désambiguïser le sens des mots polysémiques, imputer des mots manquants ou déterminer le genre lors de la traduction vers une langue ayant du genre grammatical comme avec l'anglais vers le français. Je propose deux approches principales pour intégrer la modalité visuelle: (i) un mécanisme d'attention multimodal qui apprend à prendre en compte les représentations latentes des phrases sources ainsi que les caractéristiques visuelles convolutives, (ii) une méthode qui utilise des caractéristiques visuelles globales pour amorcer les encodeurs et les décodeurs récurrents. Grâce à une évaluation automatique et humaine réalisée sur plusieurs paires de langues, les approches proposées se sont montrées bénéfiques. Enfin, je montre qu'en supprimant certaines informations linguistiques à travers la dégradation systématique des phrases sources, la véritable force des deux méthodes émerge en imputant avec succès les noms et les couleurs manquants. Elles peuvent même traduire lorsque des morceaux de phrases sources sont entièrement supprimés.

Abstract

Machine translation aims at automatically translating documents from one language to another without human intervention. With the advent of deep neural networks (DNN), neural approaches to machine translation started to dominate the field, reaching state-of-the-art performance in many languages. Neural machine translation (NMT) also revived the interest in *interlingual* machine translation due to how it naturally fits the task into an encoder-decoder framework which produces a translation by decoding a latent source representation. Combined with the architectural flexibility of DNNs, this framework paved the way for further research in multimodality with the objective of augmenting the latent representations with other modalities such as vision or speech, for example. This thesis focuses on a multimodal machine translation (MMT) framework that integrates a secondary visual modality to achieve better and visually grounded language understanding. I specifically worked with a dataset containing images and their translated descriptions, where visual context can be useful for word sense disambiguation, missing word imputation, or gender marking when translating from a language with gender-neutral nouns to one with grammatical gender system as is the case with English to French. I propose two main approaches to integrate the visual modality: (i) a multimodal attention mechanism that learns to take into account both sentence and convolutional visual representations, (ii) a method that uses global visual feature vectors to prime the sentence encoders and the decoders. Through automatic and human evaluation conducted on multiple language pairs, the proposed approaches were demonstrated to be beneficial. Finally, I further show that by systematically removing certain linguistic information from the input sentences, the true strength of both methods emerges as they successfully impute missing nouns, colors and can even translate when parts of the source sentences are completely removed.

Table of Contents

List of Tables

List of Figures

List of Notations

$\mathbf{A}$	A sequence of tokens
$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
a	A scalar
$\boldsymbol{a}$	A vector
a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
$\boldsymbol{A}$	A matrix
$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise (Hadamard) product of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
A	A tensor
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor A
$\mathsf{A}_{:,:,i}$	2-D slice of a 3-D tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$

$P(\mathrm{a})$	A probability distribution over a discrete variable
$\mathrm{a} \sim P$	Random variable a has distribution P
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\dfrac{1}{1+\exp(-x)}$
p_{data}	The data generating distribution
$\hat{p}_{\mathrm{data}}$	The empirical distribution defined by the training set
$\mathcal{D}$	A set of training examples
$\boldsymbol{x}^{(i)}$	The i-th example (input) from a dataset
$y^{(i)}$	The target label associated with $\boldsymbol{x}^{(i)}$

List of Abbreviations

AI	Artificial Intelligence
AMMT	Attentive MMT
ASR	Automatic Speech Recognition
BGD	Batch Gradient Descent
BOS	Beginning-of-sentence
BP	Backpropagation
BPE	Byte Pair Encoding
BPTT	Backpropagation Through Time
CGRU	Conditional GRU
CNN	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
E2E	End-to-End
EOS	End-of-sentence
FC	Fully-connected
FCNN	Fully-connected Neural Networks
GAP	Global Average Pooling
GRU	Gated Recurrent Unit
IC	Image Captioning
LM	Language Modeling
LSTM	Long Short-term Memory
MGD	Minibatch Gradient Descent

ML	Machine Learning
MLE	Maximum Likelihood Estimation
MMT	Multimodal Machine Translation
MT	Machine Translation
MTL	Multitask Learning
NIC	Neural Image Captioning
NLM	Neural Language Modeling
NMT	Neural Machine Translation
OOV	Out-of-vocabulary
PBMT	Phrase-based Machine Translation
RNN	Recurrent Neural Networks
S2S	Sequence-to-Sequence
SGD	Stochastic Gradient Descent
TFNMT	Transformer NMT
UNK	Unknown Token
VQA	Visual Question Answering
WSD	Word Sense Disambiguation

CHAPTER 1

Introduction

Language is the primary framework of communication that human beings use, when expressing their ideas and thoughts. The existence of thousands of languages in the world however, constitutes an obstacle to communication between the speakers of different languages. Although human translation is the gold standard for high quality translation across languages, nowadays we also require decent instantaneous translation facilities for different purposes such as quickly understanding a newly received document or making sense of a critical sign during a touristic trip. A computational solution to the instantaneous translation problem is not only important for the primary task of text translation but also is key to remove the communication barrier between speakers of different languages by means of a conversational tool that combines speech recognition, translation and speech synthesis for example. To that end, machine translation (MT) is specifically interested in automatic language translation, through the use of statistical modeling tools of machine learning (ML). These tools aim to capture the "complex relations" between two collections of sentences that are translations of each other. These complex relations mostly refer to linguistic aspects such as syntax, semantics and pragmatics which are key to language understanding. An MT model should thus be able to understand a source language and then construct a fluent and adequate translation in the target language. Until recently, the state-of-the-art approaches in MT heavily relied on multi-stage pipelines that divide the translation problem into smaller parts. These parts are primarily responsible for modeling the phrase translation probabilities, learning the most likely target-to-source word alignments and ensuring the fluency of the produced translations (Koehn et al., 2003). Nowadays, deep neural networks based approaches (Bahdanau et al., 2014; Sutskever et al., 2014; Vaswani et al., 2017) are dominating the field and considered to be the new state-of-the-art in MT. Unlike the multi-stage approach, neural MT models (NMT) are end-to-end and relatively easily trained with almost no feature engineering involved.

Regardless of the underlying statistical framework, MT requires large amount of parallel sentences to be able to learn a decent translation model. Luckily, we are in an era where massive amount of data is constantly produced and made publicly available through the Internet. The availability of such diverse data ranging from documents and images to videos, also gives rise to numerous new ideas to foster research on multimodal machine learning, a term coined to designate models that can leverage information coming from different modalities (Baltrusaitis et al., 2017). This research area is inspired by the multimodal aspects of human learning *i.e.* the inherent ability of human beings to integrate simultaneous information from different sensory channels. In infant learning for example, lexical items produced through pointing gestures were shown to later migrate to the verbal lexicon of the children (Iverson and Goldin-Meadow, 2005) whereas Abu-Zhaya et al. (2017) provide evidence that infants benefit more from tactile-speech than visual-speech interactions. The multisensory integration ability also allows us to achieve a better understanding of the surrounding world (Stein et al., 2009; Ernst and Banks, 2002) by reducing uncertainty, for example when we attempt to recognize speech in a noisy environment.

Similar uncertainties also arise in the case of MT where for example a word in a source sentence has multiple senses or when the gender information has to be inferred for translating from a gender-neutral language to another one that has grammatical gender. An example to the latter ambiguity is as follows: Translating "a basketball player" to French requires inferring the sex of the player in order to select between "un joueur" (male) and "une joueuse" (female). The primary objective of this thesis is thus to devise multimodal machine translation (MMT) systems which leverage contextual information from an auxiliary input modality. In order to do so, we explore a relatively new dataset called Multi30K (Elliott et al., 2016) which provides images, their natural language descriptions in English and the translations of these descriptions into three different languages. The choice of vision as the auxiliary modality here is motivated by the fact that the images are (almost) objective depictions of concrete concepts surrounding us, making them natural candidates to resolve the aforementioned linguistic ambiguities. Moreover, evidence from the literature also suggest their usefulness in terms of joint language and vision processing: Bergsma and Van Durme (2011) and Kiela et al. (2015) used images in a visual similarity based bilingual lexicon induction task *i.e.* the task of inferring the translation of a word without having access to data directly labeled for translation purposes; Vinyals et al. (2015) and Xu et al. (2015b) demonstrated the possibility to generate natural language descriptions for images using end-to-end deep neural networks.

To that end, I mainly propose two different interaction methods based on two different computational representations of images. Both types of features are obtained from state-of-the-art deep computer vision models (Simonyan and Zisserman, 2014; He et al., 2016) which are pre-trained to perform ImageNet large-scale image classification task (Deng et al., 2009). Before getting into the details of the proposed approaches, I first provide an extensive background about ML, especially focusing on the ecosystem around deep neural networks (Chapter 2) and the underlying details of the state-of-the-art pre-trained computer vision models (section 2.6.3, p. 27). I then describe the conventional multi-stage MT and the state-of-the-art NMT approaches in chapter 3. In chapter 4, I explain the MMT task along with the Multi30K dataset and provide a detailed literature overview of the state-of-the-art in MMT.

The second part of the thesis consists of our contributions to MMT. This part begins with the introductory chapter 5 which gives a thorough description of the common experimental framework of the thesis, including details such as the pre-processing pipeline, the baseline NMT architecture and the underlying software used to train the models. Chapter 6 and chapter 7 introduce the two family of multimodal interactions. The first family of interactions incorporate global visual features which are high-level vectoral semantic representations, while the second family integrates more sophisticated convolutional features that preserve spatial information unlike the former. We conduct an extensive set of experiments followed by quantitative analyses for English→German and English→French translation tasks of Multi30K. Finally in chapter 8, I take a step back and provide several qualitative analyses to showcase the strengths and weaknesses of the explored MMT models, along with a novel probing framework to assess the visual awareness of the models. I conclude the thesis in chapter 9 where I discuss future perspectives about MMT and multimodal language understanding in general.

CHAPTER 2

Deep Neural Networks

Machine learning (ML) is traditionally considered as a multi-stage framework which breaks down the task to be solved into two main stages. If we consider a supervised learning problem such as object recognition, the first stage – referred to as *feature engineering* – would aim at extracting useful features from raw input images while the second stage would train a classifier to estimate the probability distribution over plausible object labels given the extracted input features. This *feature engineering* stage requires a substantial amount of human expertise and domain-knowledge. In addition, the quality of the obtained features heavily affects the performance of the final model.

Deep neural networks (DNN) on the other hand, propose to transform the explicit *feature engineering* stage into an intrinsic aspect of the model referred to as *representation learning* (Goodfellow et al., 2016). DNNs are able to jointly learn sophisticated feature extractors and an output logic – to perform classification or regression for example – by minimizing a task-relevant error signal through stochastic optimization. Unlike explicit *feature engineering*, this optimization framework enable DNNs to learn good feature extractors that even humans may not be able to come up with. In contrast to multi-stage ML, DNNs are also end-to-end: they require minimum to none pre/post processing allowing them to be easily trained and deployed.

The idea behind DNNs dates back to 1950s. Initially, AI researchers were inspired by the massively interconnected network of neurons found in the biological brain. This biological evidence of intelligence guided the field to come up with simple computational units such as the McCulloch-Pitts neuron (McCulloch and Pitts, 1943) and later the perceptron algorithm (Rosenblatt, 1958). Unfortunately, the lack of efficient training algorithms and the alleged inability of these models to learn the exclusive-OR (XOR) function had triggered the so-called AI winter where research on neural networks had lost traction (Goodfellow et al., 2016). Luckily, a group of researchers continued to work

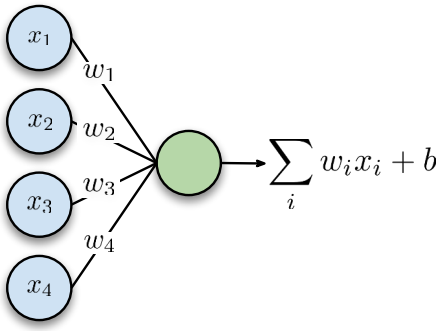


Figure 2.1: The graphical model of a neuron with four inputs: although the input neurons are not parameterized, they are generally depicted using nodes as well for notational purposes. *Blue* and *green* shades represent the input and output nodes respectively.

in the field resulting in the discovery of the missing piece of the equation, the back-propagation algorithm which is still a crucial element of DNN training (Werbos, 1974; Rumelhart et al., 1986). Today, DNNs are considered state-of-the-art in many fields including but not limited to object recognition, automatic speech recognition (ASR), language modeling (LM) and machine translation (MT) (LeCun et al., 2015).

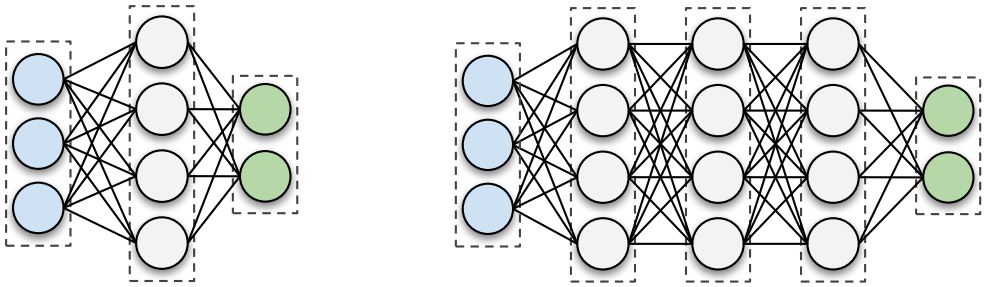
In this chapter, I will first introduce the fundamentals of DNNs with a focus on supervised learning. I will then proceed with recurrent neural networks (RNN), a type of DNN specialized for modeling sequential data such as natural languages. Finally, in order to lay the ground for joint language and vision processing, I will describe convolutional neural networks (CNN) which are state-of-the-art models in image and video processing.

2.1 Neurons and Fully-connected Networks

The basic computational unit in a DNN is a neuron. Parameterized with a set of weights $\{w_i\}_1^n$ and a bias term b, a neuron outputs the weighted sum of its inputs (Figure 2.1). The parameters of a modern neuron are real valued unlike the early McCulloch-Pitts neuron in the literature which used binary connections (McCulloch and Pitts, 1943). It is also possible to interpret the weighted sum as a *dot product* between the input vector $\boldsymbol{x} = [x_1; x_2; \dots; x_n]^\top$ and the weight vector $\boldsymbol{w} = [w_1; w_2; \dots; w_n]^\top$ as follows:

$$\hat{y} = \sum_i w_i x_i + b = \boldsymbol{x}^\top \boldsymbol{w} + b \tag{2.1}$$

A neuron learns to produce a real valued response to some particular input pattern where the response is proportional to the angular distance (*i.e.* closeness) between the input and the learned weight vector. This particular view of the neuron as a pattern detector hints at the fact that, analogous to biological brain, complex reasoning ability



(a) Single layer fully-connected network (b) 3-layer fully-connected network

Figure 2.2: Fully-connected networks with one and three hidden layers: The naming convention only reflects *the number of hidden layers* as in "single layer" and "3-layer".

may be achieved through an interconnected network of neurons – *i.e. neural networks*. Before getting familiar with the concept of *neural networks* however, we need to define one more abstraction, namely, *a layer*, which is a logical computation unit grouping a set of neurons. The fundamental layer type in modern DNNs is the *fully-connected* layer (FC) which consists of h neurons, each connected to the incoming layer with dedicated weight vectors. The weight vector in equation 2.1 can be replaced with a matrix $\boldsymbol{W}$ where the i-th row corresponds to the weight vector of the i-th neuron in the layer. This way, the output of the layer becomes a vector $\hat{\boldsymbol{y}}$ given by a matrix-vector product:

$$\hat{\boldsymbol{y}} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \qquad (\boldsymbol{W} \in \mathbb{R}^{h \times n}, \boldsymbol{x} \in \mathbb{R}^{n}, \boldsymbol{b} \in \mathbb{R}^{h}, \hat{\boldsymbol{y}} \in \mathbb{R}^{h}) \tag{2.2}$$

We can now define a *neural network* (NN) as an interconnected topology made of *input layers*, *output layers* and *hidden layers* stacked in-between them. The latter layers are called *hidden* as their outcomes are not observable from the actual data generating processes *i.e.* they are considered to be variables expected to model latent structures discovered from the input. We will be using the term fully-connected neural networks (FCNN) to refer to networks that consist of FC layers. When naming FCNNs, the convention ignores the enumeration of the input and the output layers and only counts the number of hidden layers in-between. Figure 2.2 shows two FCNNs, a single-layer and a three-layer one, where the hidden layer neurons are shaded with *gray*. Let us express the computation performed by the first FCNN where the output of the network is computed by successively feeding the output of each previous layer as input to the next one. $\ell_i()$ denotes the function of the i-th layer where the parameters are $\boldsymbol{W}^{(i)}$ and $\boldsymbol{b}^{(i)}$:

$$\begin{aligned} \hat{\boldsymbol{y}} &= \mathrm{FCNN}_1(\boldsymbol{x}) = \ell_2(\ell_1(\boldsymbol{x})) \\ &= \boldsymbol{W}^{(2)}\left(\ell_2(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}\right) \end{aligned} \tag{2.3}$$

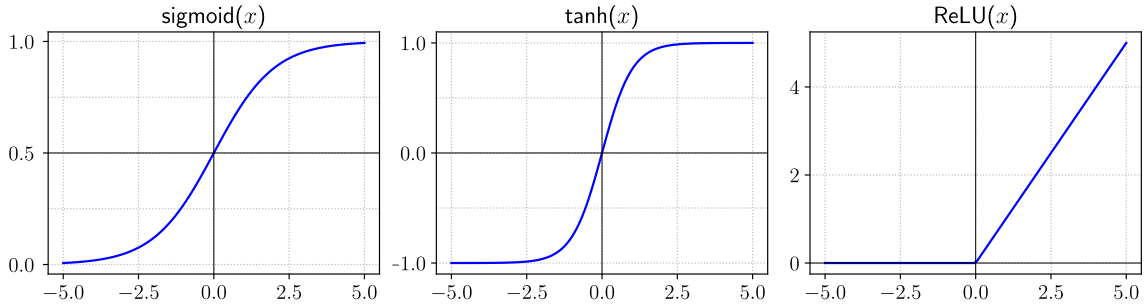


Figure 2.3: Commonly used non-linear activation functions.

Non-linear Neurons

So far, we have only covered linear layers where each neuron basically computes a different linear combination of the incoming connections. Although increasing the number of hidden layers seem to add computational capacity to the network, linear models are not able to capture non-linear input-output mappings, a traditional example being the XOR function. On the other hand, it has been shown that shallow FCNNs can act as "universal function approximators" once equipped with sigmoid non-linearities (Cybenko, 1989). For these reasons, modern DNNs are inherently designed with non-linear activation functions which themselves constitute an active area of research (Glorot and Bengio, 2010; Xu et al., 2015a; Clevert et al., 2015; Klambauer et al., 2017).

The three most commonly used non-linear activation functions are plotted in Figure 2.3. Sigmoid activations are generally used to implement gating mechanisms in DNNs that regulate the information flow (section 2.5.1, p. 20). Tanh and ReLU activations are more general purpose and often used within RNNs (section 2.5, p. 18) and CNNs (section 2.6, p. 24) to induce complex pattern recognition abilities. These functions are mathematically defined as follows:

$$\begin{aligned}
\text{sigmoid}(x) = \sigma(x) &= \frac{1}{1 + \exp(-x)} \\
\tanh(x) &= 2\sigma(2x) - 1 \\
\text{ReLU}(x) &= \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}
\end{aligned}$$

The application of an activation function $\phi : \mathbb{R} \mapsto \mathbb{R}$ to a vector implies that it is applied to each component of that vector. The following depicts the three layer FCNN (Figure 2.2b) by assigning a non-linearity $\phi_i()$ to each layer:

$$\hat{\boldsymbol{y}} = \text{FCNN}_2(\boldsymbol{x}) = \phi_4(\ell_4(\phi_3(\ell_3(\phi_2(\ell_2(\phi_1(\ell_1(\boldsymbol{x}))))))))$$

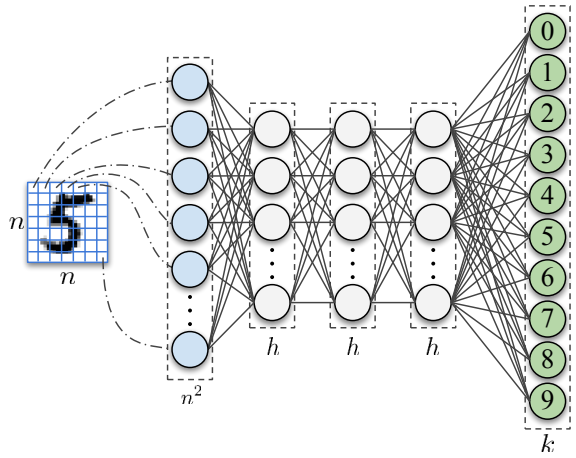


Figure 2.4: A simple FCNN for handwritten digit recognition: the *dashed* arrows on the left part indicate that the actual input is a flattened version of the 2D input image.

2.2 Multi-class Classification

Before diving into more sophisticated types of layers and networks, let us introduce the classical handwritten digit recognition to illustrate the steps involved in supervised training of a neural network. Figure 2.4 proposes a simple three-layer FCNN in order to estimate the probability distribution over a set of *labels*, given an input image. The model receives a flattened vector $\boldsymbol{x} \in \mathbb{R}^{n^2}$ representing a grayscale square input image of shape $n \times n$, feeds it through the subsequent hidden layers of size h each and produces a vector of predicted probabilities $\hat{\boldsymbol{y}} \in \mathbb{R}^k$. The set of digit labels is defined as $K = \{0, 1, \ldots, 9\}$ and the number of labels is given by the cardinality of the label set *i.e.* $k = |K| = 10$.

A well known dataset for handwritten digit recognition is the MNIST dataset (LeCun et al., 1998) which provides 60K training and 10K testing examples. We denote the training set by $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right) : 1 \leq i \leq N\right\}$ where each element is an ordered pair of one flattened image vector $\boldsymbol{x}^{(i)}$ and its target label $y^{(i)} \in K$. Since the images provided by MNIST are of shape 28×28, the size of a flattened image vector is $28 \times 28 = 784$.

Both the digit recognition task and the various NMT models that will be explained in future sections, perform a *multi-class classification i.e.* predicts a discrete categorical distribution over a predefined set of labels. A linear neuron produces an unbounded response which is obviously not what we expect from the output layer of such models. Instead, we would like that the output produces a valid probability distribution. We achieve this by using a special operator *softmax* $\left(\mathbb{R}^k \mapsto [0, 1]^k\right)$ which normalizes its

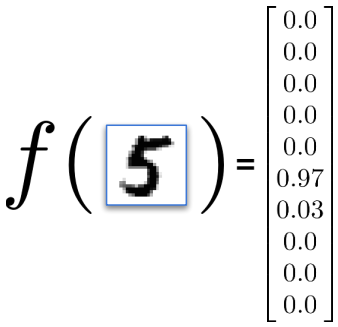


Figure 2.5: High-level abstraction of the forward-pass step in a DNN: given an input image, the network assigns probabilities of 0.97 and 0.03 to labels 5 and 6 respectively.

input vector so that the values lie between $[0, 1]$ and sum to 1:

$$\text{SOFTMAX}\,(\boldsymbol{z}) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k}\exp(z_i)}; \frac{\exp(z_2)}{\sum_{i=1}^{k}\exp(z_i)}; \ldots ; \frac{\exp(z_k)}{\sum_{i=1}^{k}\exp(z_i)}\right]$$

Denoting the network by a function $f : \mathbb{R}^{n^2} \mapsto \mathbb{R}^k$ and setting the output layer activation to softmax, a *forward-pass* through the network (Figure 2.5) can now predict $P(\boldsymbol{y} \mid \boldsymbol{x})$ *i.e.* the conditional probability distribution over the labels given an image. For example, the last equation below fetches the probability of the input being a "5":

$$\begin{aligned} P(\boldsymbol{y} \mid \boldsymbol{x}) = \hat{\boldsymbol{y}} &= f(\boldsymbol{x}) \\ &= \text{SOFTMAX}\,(\ell_4(\phi_3(\ell_3(\phi_2(\ell_2(\phi_1(\ell_1(\boldsymbol{x}))))))))\\ P(y = \text{“5”} \mid \boldsymbol{x}) = \hat{y}_5 &= 0.97 \end{aligned} \tag{2.4}$$

2.3 Maximum Likelihood Estimation

The training set $\mathcal{D}$ is just a sample from the true data generating distribution p_{data}, which is what we actually want to understand in order to perform inference later on using unseen data. A common framework to achieve this is the maximum likelihood estimation (MLE) where the objective is to find a set of parameters that maximize the likelihood of the training set, or to put it differently maximize the probability of the ground-truth label assigned by the model. In order to cast this as an optimization problem, we first need to pick a loss function suitable for multi-class classification. A common choice is *negative log-likelihood* (NLL) which is defined below for a single example $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$:

$$\text{NLL}^{(i)} = -\log\left(P\left(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)\right) = -\log\left(\hat{y}_{y^{(i)}}\right)$$

Note that the explicit $\boldsymbol{\theta}$ states that the model is parameterized by $\boldsymbol{\theta}$ which is a flattened parameter vector containing all the weights and the biases of the model. We can now define the training set NLL as the expected loss over all the examples:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \text{NLL}^{(i)} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(P\left(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)\right)$$

As can be seen, NLL is a natural choice for classification since it approaches 0 when the output probability for the correct label approaches 1 and slowly goes to infinity otherwise. This way, we can cast MLE as minimizing the training NLL over the parameter space where the final parameter estimate is denoted by $\boldsymbol{\theta}^*$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{i=1}^{N} \log\left(P\left(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)\right)$$

2.4 Training DNNs

One nice property of the network depicted so far is its compositional nature: each layer in the topology is a function of its inputs parameterized with the weights and biases of that layer. This means that the final NLL loss is differentiable with respect to all parameters involved in the network *i.e.* the parameter vector $\boldsymbol{\theta}$. When equipped with the necessary mathematical tools, the differentiable nature of the network allows one to compute the gradient of the loss function with respect to $\boldsymbol{\theta}$ denoted by $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$. This gradient vector – composed of partial derivatives – quantifies how much the loss function changes in response to an infinitely small change in each parameter θ_i. The following shows how the gradient vector is defined for a network with D parameters *i.e.* $\boldsymbol{\theta} \in \mathbb{R}^D$:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \begin{bmatrix} \dfrac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_1} \\ \vdots \\ \dfrac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_D} \end{bmatrix} \tag{2.5}$$

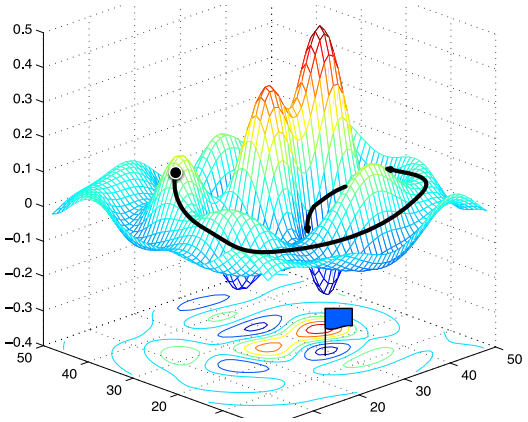


Figure 2.6: The loss surface[1] of a function with two parameters: gradient descent allows going downhill from the initial point (•) to a local minimum (blue flag).

Since the gradient vector points in the direction of greatest rate of increase and our objective is to minimize the loss, we can update $\boldsymbol{\theta}$ by taking steps towards the *negative* of the gradient vector to decrease the loss (Figure 2.6). This iterative optimization method is called *batch gradient descent* (BGD) and it forms the basis of modern DNNs (Rumelhart et al., 1986; LeCun et al., 1998). The described update rule is given as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{2.6}$$

The scalar hyperparameter α is called the *learning rate* which tunes the size of the steps taken during the update rule. Correctly setting the learning rate is of utmost importance since a too small learning rate can lead to slow convergence while a large one may provoke oscillations around local minima preventing convergence.

2.4.1 Minibatch Gradient Descent

Although the update rule for BGD (Equation 2.6) computes the gradient of the entire training set loss $\mathcal{L}(\boldsymbol{\theta})$ with respect to the parameters, in reality we prefer to split the training set into smaller chunks called *minibatches* and use the gradient of the minibatch loss during training. This approach called *minibatch gradient descent* (MGD) has mainly two advantages over BGD: (i) it increases the number of parameter updates performed in a single sweep of the training set allowing a detailed exploration of the parameter space and (ii) it makes it possible to efficiently train a model over datasets of hundreds of thousands and even millions of training examples. The latter efficiency is due to the fact that CPUs and GPUs are highly tuned for batched linear algebra operations.

[1]Illustration adapted from Huang et al. (2017a) with permission.

To sum up, let us denote the number of samples in a minibatch by B. It is trivial to see that by setting B equal to the size of the training set, MGD reduces to BGD. On the other hand, setting $B{=}1$ leads to the *online* BGD called *stochastic gradient descent* (SGD). SGD traverses the training set one example at a time and applies the update rule after each such example. Although this is rarely used in practice because of its computational inefficiency, the term SGD often appears in the literature to actually refer to MGD.

2.4.2 Adaptive Optimizers

Several adaptive extensions to gradient descent have been proposed in the last decade to integrate feature specific learning rate scheduling (Duchi et al., 2011; Zeiler, 2012; Kingma and Ba, 2014; Reddi et al., 2018). The common idea behind these methods is to store the statistics of previous gradients (and possibly their magnitudes) and use their running averages to accelerate or slow down per-feature learning. Nowadays, these adaptive methods are generally the starting point for researchers and practitioners as they offer very good out-of-the-box performance, which is the reason I used an adaptive algorithm called ADAM (Kingma and Ba, 2014) throughout the experiments in this thesis. Using ADAM, the new parameter vector $\boldsymbol{\theta}_t$ at timestep t is obtained as follows:

$$\begin{aligned}
\boldsymbol{g}_t &\leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1}) \\
\boldsymbol{m}_t &\leftarrow (1-\beta_1)\boldsymbol{g}_t + \beta_1 \boldsymbol{m}_{t-1} \\
\boldsymbol{v}_t &\leftarrow (1-\beta_2)\boldsymbol{g}_t{}^2 + \beta_2 \boldsymbol{v}_{t-1} \\
\boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \frac{\alpha}{\sqrt{\boldsymbol{v}_t}} \boldsymbol{m}_t
\end{aligned} \tag{2.7}$$

In the above, $\boldsymbol{g}_t$ is a shorthand for the gradient vector while $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ are the exponential moving averages of the gradient and the squared gradient vectors (with decay rates β_1 and β_2). We can see from equation 2.7 that the base learning rate α is now scaled using $\sqrt{\boldsymbol{v}_t}$ and the actual gradient $\boldsymbol{g}_t$ is replaced with an exponential moving average $\boldsymbol{m}_t$.

2.4.3 Parameter Initialization

The training starts by randomly sampling an initial $\boldsymbol{\theta}$ vector through a procedure called *parameter initialization*, an active area of research itself (Martens, 2010; Glorot and Bengio, 2010; Saxe et al., 2013; He et al., 2015; Arpit and Bengio, 2019). Failing to initialize the parameters correctly is likely to hinder the training process by causing slow convergence or even no convergence at all. The parameter initialization is even more important in DNNs with non-linear activation functions (Section 2.1) since incorrect initialization can cause neurons to saturate *i.e.* staying in a constant regime which propagates back a

zero gradient that inhibits learning (Goodfellow et al., 2016). In the following, we make use of the initialization method proposed by He et al. (2015) where the variance of the sampled weights for a layer with H inputs is scaled by $\sqrt{2/H}$. This per-layer standard deviation makes sure that the variance of layer activations are preserved throughout the depth of the network. We specifically sample the weights from the following gaussian distribution $\mathcal{N}(0; \sqrt{2/H})$.

2.4.4 Regularization

So far, we have shown how to formulate the training problem from an optimization point of view. Although it may be intuitive to think that the overall aim of the minimization framework is to estimate a parameter vector $\boldsymbol{\theta}^*$ which obtains $\sim$0 loss, this is hardly what we would like to achieve. More precisely, such models perfectly memorizing (overfitting) the training set will exhibit poor performance on a held-out test set *i.e.* they will not generalize well to unseen samples. Ideally, what we would like to end up with is a model which achieves a small training loss as well as a small gap between this training loss and the test set loss. The violation of these principles are referred to as underfitting and overfitting (Goodfellow et al., 2016). The overfitting can be mitigated by carefully regularizing the capacity of the model to ensure the *law of parsimony i.e.* to encourage simpler solutions over very complex ones. On the other hand, underfitting – when not caused by aggressive regularization – generally requires increasing the explicit capacity of the model defined by the width, the depth and the types of layers in the case of a DNN. In what follows, I describe three commonly used regularization techniques.

L_2 Regularization

One classical way of regularization is the so called L_2 penalty which is additively combined with the training loss to be minimized. Let us redefine the loss function as a sum of the previously introduced training NLL and the L_2 penalty term and denote it by $\mathcal{J}$:

$$\begin{aligned} \mathcal{J}(\boldsymbol{\theta}) &= \mathcal{L}(\boldsymbol{\theta}) + \lambda \, ||\boldsymbol{\theta}||_2^2 \\ &= \mathcal{L}(\boldsymbol{\theta}) + \lambda \sum_i^D \theta_i^2 \end{aligned} \tag{2.8}$$

This penalty term scaled with λ imposes a constraint over the parameter space such that the L_2 norm of the parameter vector[2] is minimal *i.e.* an arbitrary subset of weights is discouraged to become very large unless it is necessary (Krogh and Hertz, 1992). In

[2]In general, L_2 penalty term is not applied to biases (Goodfellow et al., 2016).

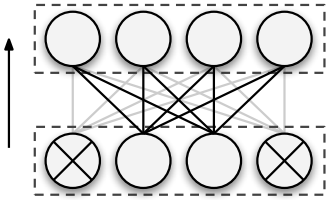


Figure 2.7: A fully-connected layer with dropout regularization: bottom layer drops out half of its activations which is equivalent to multiplying them by zero.

other words, the penalty term encourages cooperation rather than relying on a set of neurons with large weights prone to capture features not necessarily useful towards generalization or even noise patterns. L_2 regularization is generally used interchangeably with weight decay although the latter explicitly appears in the update rule (equation 2.6) while the former penalizes the loss as in equation 2.8 (Loshchilov and Hutter, 2019).

Dropout

Another regularization technique pervasively used throughout the literature is the so-called *dropout* (Srivastava et al., 2014), which when applied to a layer, stochastically samples a subset of the activations with a predefined probability and multiplies them by zero (Figure 2.7). This procedure which is repeated for each batch during training, has the effect of training exponentially many partially-connected networks that are optimized through the same objective function. The stochastic removal of hidden units prevents upper layers from becoming "lazy" *i.e.* relying on the constant availability of some highly predictive incoming states. When the model has to be used in evaluation mode, the dropout functionality is removed and the activations to the post-dropout layer are correspondingly scaled to match the expected incoming magnitude. Although there are advanced dropout variants especially suited for recurrent neural networks (Gal and Ghahramani, 2016; Semeniuta et al., 2016), the simple approach is quite effective in-between non-recurrent layers such as fully-connected and convolutional ones.

Early-Stopping

The final regularization technique that I would like to mention is *early-stopping*. The idea here is to periodically evaluate the performance of the model on a special validation set and save the parameters if the performance improves over the previous best model. If there is no improvement for a predetermined amount of time (patience), the training is stopped and the last saved parameters are considered as the final ones. Early-stopping

thus avoids overfitted models by returning back in time to the model with the best generalization ability. When dealing with language related tasks, we will often see the interplay of the empirical loss $\mathcal{L}$ that the training minimizes with a task-specific performance metric that can for example quantify "how good a translated sentence is". Although we may be more curious about the latter, these metrics are generally not differentiable with respect to the parameters; hence the reason why we choose to minimize the empirical loss instead. Early-stopping also gives us the ability to use such task-specific metrics in order to assess how well a model is doing.

2.4.5 Backpropagation

We previously saw that given an arbitrary input, the loss is computed by what we call a forward-pass through the network *i.e.* a successive application of functions defined in the topology. We also know that each parameter will be accordingly updated with respect to its partial derivative $\frac{\partial \mathcal{J}}{\partial \theta_i}$. The missing piece in the overall training algorithm is the middle step which will compute those partial derivatives. In the context of neural networks, this step is achieved by the *backpropagation* (BP) algorithm for which an efficient formulation was first proposed by Werbos (1982) and later popularized by Rumelhart et al. (1986); LeCun (1988) according to Schmidhuber (2015).

BP is essentially a special case of *reverse-mode automatic differentiation* (RAD) that propagates the scalar loss signal backward in order to compute the partial derivatives (Baydin et al., 2017). When doing so, it defines the overall function that the network computes in terms of smaller building blocks such as variables and operators (multiplication, addition, trigonometric functions, etc.). Each such building block (node) has well defined forward/backward semantics that define the forward computation and backward gradient propagation scheme. During the forward-pass, each node stores intermediate results and keeps track of its dependencies while the backward-pass reuses those intermediate results and neatly propagates back the gradients into the necessary nodes. When a scalar loss function is used – typically the case with many DNN models – the time complexity of the forward and the backward propagations are almost the same (Baydin et al., 2017).

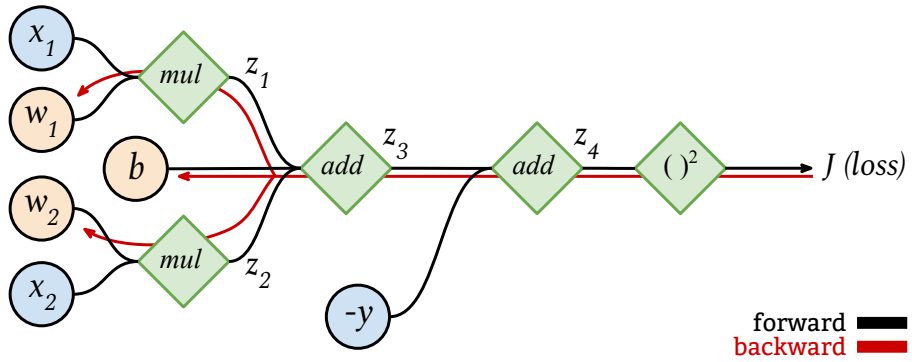


Figure 2.8: Computation graph of a simple linear regression model: y is the ground-truth value for this specific input (x_1, x_2) while the parameters are $\{w_1, w_2, b\}$.

To concretize BP, let us give a toy example that illustrates a linear regression model with quadratic error (Figure 2.8). We define a set of intermediate variables z_i's (left) and write down their partial derivatives with respect to their inputs (right):

$$
\begin{aligned}
J &= ((w_1 x_1 + w_2 x_2 + b) - y)^2 = {z_4}^2 & \partial J / \partial z_4 &= 2 z_4 \\
z_4 &= z_3 - y & \partial z_4 / \partial z_3 &= 1 \\
z_3 &= z_1 + z_2 + b & \partial z_3 / \partial z_1 = \partial z_3 / \partial z_2 = \partial z_3 / \partial b &= 1 \\
z_2 &= w_2 x_2 & \partial z_2 / \partial w_2 &= x_2 \\
z_1 &= w_1 x_1 & \partial z_1 / \partial w_1 &= x_1
\end{aligned}
$$

Once we compute the gradient of the loss J with respect to the model parameters using the chain rule, we clearly see that they are compositionally made up of intermediate gradient expressions (blue). Each parameter then receives its gradient after the error is propagated back towards the inner parts of the network:

$$
\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial z_4} \frac{\partial z_4}{\partial z_3} \frac{\partial z_3}{\partial z_1} \frac{\partial z_1}{\partial w_1} = 2 z_4 x_1
$$

$$
\frac{\partial J}{\partial w_2} = \frac{\partial J}{\partial z_4} \frac{\partial z_4}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial w_2} = 2 z_4 x_2
$$

$$
\frac{\partial J}{\partial b} = \frac{\partial J}{\partial z_4} \frac{\partial z_4}{\partial z_3} \frac{\partial z_3}{\partial b} = 2 z_4
$$

Vanishing and Exploding Gradients

Depending on the depth of the network topology and the layer types, the magnitude of the gradient vector can become very small (vanishing) or very large (exploding) during training. The former may eventually hinder the learning for layers that receive very small

Initialize $\boldsymbol{\theta}$ randomly;
patience $\leftarrow P$;
while *patience* > 0 **do**
  // An epoch consumes $\mathcal{D}$
  foreach *minibatch in* $\mathcal{D}$ **do**
    $\mathcal{J} \leftarrow \mathcal{L}(\text{batch}; \boldsymbol{\theta}) + \lambda \, ||\boldsymbol{\theta}||_2^2$; // forward-pass
    Compute $\nabla_{\boldsymbol{\theta}} \mathcal{J}$; // backward-pass
    Update $\boldsymbol{\theta}$ through the optimizer of choice;
  end
  if $\mathcal{L}(\mathcal{D}_{\text{valid}}; \boldsymbol{\theta})$ *is the best so far* **then**
    save model parameters $\boldsymbol{\theta}$;
  else
    patience $\leftarrow$ patience - 1;
  end
end

Figure 2.9: The complete training algorithm with L_2 regularization and early-stopping

gradients while the latter is bad for numerical stability. A common technique to mitigate exploding gradients is to apply *gradient clipping* (Pascanu et al., 2013) to renormalize the magnitude of the gradient vector if its norm is higher than a predetermined threshold.

The vanishing gradient problem is more of an issue in very deep CNNs and RNNs that will be depicted in the following sections. In both cases residual connections (He et al., 2016) from the bottom layers to the top of the network are generally helpful to create auxiliary pathways for the gradients to backpropagate. For RNNs, advanced units with gating mechanisms (Hochreiter and Schmidhuber, 1997; Cho et al., 2014b) are *de facto* preferred over the original recurrent units (section 2.5.1, p. 20).

2.4.6 The Complete Algorithm

Now that we have all the fundamental pieces covered, we can formalize the overall training process as a well defined algorithm (Figure 2.9). Once we have a neural architecture decided, the training starts by randomly initializing the parameter vector $\boldsymbol{\theta}$ and setting some other hyperparameters such as the early-stopping patience. A full sweep over the training set is referred to as an *epoch* which is itself randomly divided into minibatches of examples. An iteration consists of performing the forward-pass, the backward-pass and the parameter update over a single minibatch. In order to do early-stopping, the generalization performance of the model is periodically assessed over a held-out validation set $\mathcal{D}_{\text{valid}}$. The performance criterion here does not necessarily have to be the NLL loss

used as the training objective but can be some other task-relevant metrics such as translation quality or accuracy. The period of the evaluation is also a matter of choice that depends on the task and the size of the training set: it can range from some thousands of minibatches to one epoch or two. Finally, the training is stopped if no performance improvement occurs over the previously saved model after P consecutive evaluations.

2.5 Recurrent Neural Networks

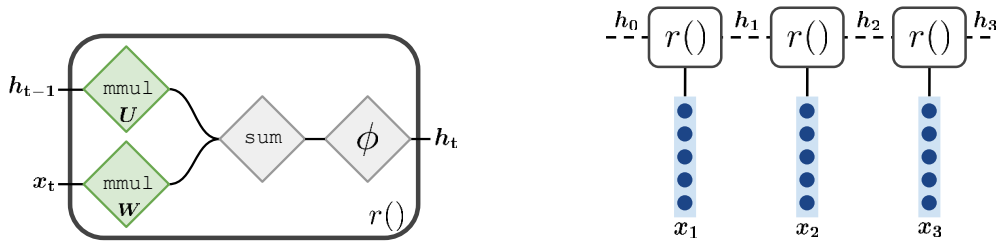
In this section, I will describe the prominent DNN type in sequential modeling, namely, recurrent neural networks (RNN). RNNs are extensively used in language related tasks such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2014; Johnson et al., 2016), image captioning (Xu et al., 2015b) and speech recognition (Chan et al., 2016). RNNs (Elman, 1990; Hochreiter and Schmidhuber, 1997; Cho et al., 2014b) sequentially update their hidden state as a function of the previous hidden state and the newly presented observation. The hidden state can be thought as a progressive memory that learns how to compress the input into an efficient latent representation. The stateful processing turns out to be important to handle natural language sentences where, driven by a set of well defined syntactic rules, the order of the words matters for correct and unambiguous semantics. RNNs also naturally fit into the framework of language processing since recurrent processing easily accomodates variable-length sentences.

Let us denote an input sequence by $\mathbf{X}=[\boldsymbol{x_1}, \ldots, \boldsymbol{x_t}, \ldots, \boldsymbol{x_T}]$[3] such that each element is a vector $\boldsymbol{x_t} \in \mathbb{R}^{D_X}$ representing a word. In the following, $r()$ denotes the parameterized function associated with the vanilla RNN where the parameters are the bias $\boldsymbol{b} \in \mathbb{R}^{D_H}$ and the matrices $\{\boldsymbol{W} \in \mathbb{R}^{D_H \times D_H}, \boldsymbol{U} \in \mathbb{R}^{D_H \times D_X}\}$. $r()$ computes the hidden state $\boldsymbol{h_t} \in \mathbb{R}^{D_H}$ as follows (Figure 2.10a):

$$\boldsymbol{h_t} = r(\boldsymbol{h_{t-1}}, \boldsymbol{x_t}) = \phi(\boldsymbol{W}\boldsymbol{h_{t-1}} + \boldsymbol{U}\boldsymbol{x_t} + \boldsymbol{b}) \tag{2.9}$$

A common choice for the non-linearity ϕ is the "tanh" function (Section 2.1, p. 5). The *initial state* $\boldsymbol{h_0}$ can be set to $\mathbf{0}$ or to an auxiliary feature vector that we would like the model to consider as an *a priori* information. The successive application of $r()$ to the input sequence $\mathbf{X}$ can be *serialized* by repeating the computation graph of $r()$ along the time axis. Since the same parameterized function $r()$ is reused along the time axis, the number of parameters in an RNN does not depend on the sequence length. An example of the *unfolded* view is given in figure 2.10b with a short input sequence $\mathbf{X}=[\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}]$.

[3] Note that the subscripts are in **bold** here compared to the notation $\boldsymbol{x}_t$ previously used to denote the t-th element of a vector.



(a) The basic RNN. (b) The basic RNN unfolded for three timesteps.

Figure 2.10: A vanilla RNN and its unfolded view: in the right figure, the hidden states are shown with *dashed* lines. `mmul` signifies matrix multiplication.

Unfolding the graph leads to the following equation for the final hidden state:

$$\boldsymbol{h}_3 = r(r(r(\boldsymbol{h_0}, \boldsymbol{x_1}), \boldsymbol{x_2}), \boldsymbol{x_3})$$

More often, we may want to access to all of the hidden states computed throughout the recurrence. Let us introduce a high-level computational block $\mathsf{RNN}()$ which, given the input and the initial hidden state, returns all of the hidden states. This sequence of hidden states $\mathbf{H}$ is usually referred to as *encodings* (or *annotations*) hence the function $\mathsf{RNN}()$ itself an *encoder*. Various sentence representations can be derived from $\mathbf{H}$ if one would like to "summarize" the semantics using a single vector:

$$\begin{aligned}
\mathbf{H} &= [\boldsymbol{h_1}, \ldots, \boldsymbol{h_T}] = \mathsf{RNN}\left(\mathbf{X}, \boldsymbol{h_0}\right) && \text{ENCODE} \\
\mathrm{H}_{-1} &= \boldsymbol{h_T} && \text{GET LAST STATE} \\
\mathrm{H}_{\mathrm{MAX}} &= \mathsf{MAXP}\left(\mathbf{H}\right) && \text{GET MAX-POOLED STATE} \\
\mathrm{H}_{\mathrm{AVG}} &= \tfrac{1}{T}\sum \boldsymbol{h_t} && \text{GET AVG-POOLED STATE}
\end{aligned}$$

Illustrative Example. Let us assume that we are given a hypothetical task of partially translating a sentence from one language to another. In order to cast the problem as classification over a predetermined set of words in the target language, let us further consider that partial translation in this context refers to predicting only the first word of the target sentence. We can now construct a simple architecture with an RNN encoder that compresses the input sentence into a vector which is then used for the classification:

$$\begin{aligned}
\mathbf{H} &= \mathsf{RNN}\left(\mathbf{X}, \boldsymbol{h_0}\right) && \text{ENCODE} \\
\hat{\boldsymbol{y}} &= \mathsf{SOFTMAX}\left(\boldsymbol{V}\,\mathrm{H}_{-1} + \boldsymbol{b}_v\right) && \text{CLASSIFY LAST STATE}
\end{aligned}$$

The output layer here is parameterized with $\{\boldsymbol{V} \in \mathbb{R}^{|K| \times D_H}, \boldsymbol{b}_v \in \mathbb{R}^{|K|}\}$ where K denotes the set of possible target words that we consider for the classification.

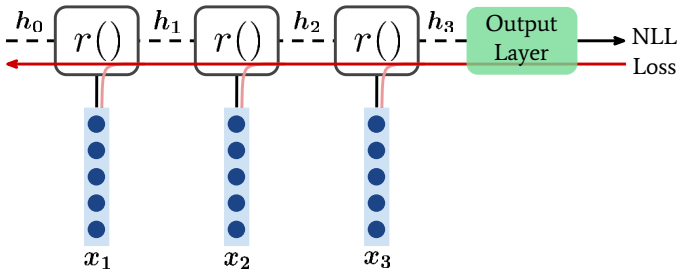


Figure 2.11: Backpropagation Through Time: the error backpropagates to each timestep (red). If $\boldsymbol{x_t}$'s are parameterized, the gradients will also flow towards them (bright red).

Backpropagation Through Time (BPTT). RNNs are trained using the previously described backpropagation algorithm as well (Section 2.4.5, p. 15) with the only difference that the error will now backpropagate through the recurrent function $r()$: the parameters of the RNN will now accumulate gradients across time since they are successively involved in the computation of all recurrent hidden states (Figure 2.11).

2.5.1 Gated RNNs

Language often involves *distant dependencies* in the form of *anaphoras*[4] or *co-references*[5] to same entities for example. Moreover, tasks such as question answering and dialog modeling further increase the span of the dependencies towards sentence and even paragraph boundaries. Although vanilla RNNs are capable of storing complex contextual informations about the input, they face difficulties when modeling dependencies between an early input $\boldsymbol{x_{t'}}$ and a late hidden state $\boldsymbol{h_t}$ where $t' << t$. These difficulties are mostly attributed to instabilities during BPTT that cause gradients to vanish (section 2.4.5, p. 16) (Bengio et al., 1994; Hochreiter, 1998). Gated RNNs incorporate sigmoid-activated gate mechanisms that dynamically regulate the information flow from the input to the hidden states as well as between successive hidden states. By doing so, they can learn to explicitly forget part of the signal or to remember it for an appropriate amount of time. The additive integration of previous states into current ones (equations 2.10 and 2.12) allows the gradient to backpropagate through distant timesteps without vanishing (Jozefowicz et al., 2015).

[4] The music was so loud that it could not be enjoyed.

[5] **I** like this book a lot because it provides an introduction to some concepts that **my** thesis will be based on," **she** replied.

Long Short-Term Memory

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is the most popular gated RNN which in turn gave rise to several further variants. LSTMs have three gates and maintain an internal *cell state* in addition to the existing *hidden state*. At timestep $\mathbf{t}$, the following computations are performed to obtain the hidden state $\boldsymbol{h_t}$:

$$
\begin{aligned}
\boldsymbol{i_t} &= \sigma(\boldsymbol{W}_i\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_i\,\boldsymbol{x_t} + \boldsymbol{b}_i) && \text{INPUT GATE} \\
\boldsymbol{f_t} &= \sigma(\boldsymbol{W}_f\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_f\,\boldsymbol{x_t} + \boldsymbol{b}_f) && \text{FORGET GATE} \\
\boldsymbol{o_t} &= \sigma(\boldsymbol{W}_o\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_o\,\boldsymbol{x_t} + \boldsymbol{b}_o) && \text{OUTPUT GATE} \\
\widetilde{\boldsymbol{c}}_\mathbf{t} &= \phi(\boldsymbol{W}_c\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_c\,\boldsymbol{x_t} + \boldsymbol{b}_c) && \text{CANDIDATE CELL STATE} \\
\boldsymbol{c_t} &= \widetilde{\boldsymbol{c}}_\mathbf{t} \odot \boldsymbol{i_t} + \boldsymbol{c_{t-1}} \odot \boldsymbol{f_t} && \text{CELL STATE} \qquad (2.10) \\
\boldsymbol{h_t} &= \phi(\boldsymbol{c_t}) \odot \boldsymbol{o_t} && \text{HIDDEN STATE} \qquad (2.11)
\end{aligned}
$$

$\odot$ denotes element-wise multiplication while σ and ϕ correspond to sigmoid and tanh non-linearities respectively. Note that the vanilla RNN is exactly recovered by setting $\boldsymbol{i_t} = \boldsymbol{o_t} = \mathbf{1}$, $\boldsymbol{f_t} = \mathbf{0}$ and by removing the non-linearity from equation 2.11.

Gated Recurrent Unit

Gated Recurrent Unit (GRU) (Cho et al., 2014b) is an LSTM variant which removes the auxiliary *cell state* and fuses the three gates into two, namely, *update* and *reset* gates:

$$
\begin{aligned}
\boldsymbol{z_t} &= \sigma(\boldsymbol{W}_z\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_z\,\boldsymbol{x_t} + \boldsymbol{b}_z) && \text{UPDATE GATE} \\
\boldsymbol{r_t} &= \sigma(\boldsymbol{W}_r\,\boldsymbol{h_{t-1}} + \boldsymbol{U}_r\,\boldsymbol{x_t} + \boldsymbol{b}_r) && \text{RESET GATE} \\
\widetilde{\boldsymbol{h}}_\mathbf{t} &= \phi(\boldsymbol{W}_h\,(\boldsymbol{h_{t-1}} \odot \boldsymbol{r_t}) + \boldsymbol{U}_h\,\boldsymbol{x_t} + \boldsymbol{b}_h) && \text{CANDIDATE HIDDEN STATE} \\
\boldsymbol{h_t} &= \widetilde{\boldsymbol{h}}_\mathbf{t} \odot \boldsymbol{z_t} + \boldsymbol{h_{t-1}} \odot (1 - \boldsymbol{z_t}) && \text{HIDDEN STATE} \qquad (2.12)
\end{aligned}
$$

When compared to LSTMs, GRUs obtain very similar performances in many sequential modeling tasks but with slightly less parameters (Chung et al., 2014; Greff et al., 2015; Jozefowicz et al., 2015). First neural approaches to machine translation incorporated both LSTMs (Sutskever et al., 2014) and GRUs (Bahdanau et al., 2014).

2.5.2 Continuous Word Representations

In section 2.5, we assumed *vectorial* word representations as inputs to RNNs but did not describe their precise nature. A naive way of representing words as vectors is the *one-hot* encoding which assigns the canonical basis vector $\boldsymbol{e}_i = [0, \ldots, 1, \ldots, 0] \in \{0, 1\}^{|K|}$

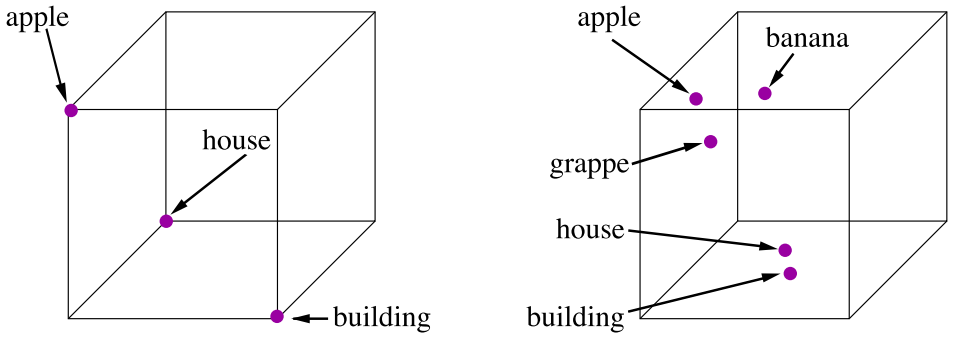


Figure 2.12: One-hot (left) vs distributional (right) word representations[6] .

to the i-th word in the vocabulary where the size of the vocabulary is $|K|$. Figure 2.12 (left) shows a 3D space with three one-hot encoded words. It can be easily seen that this approach does not encode the notion of word similarity at all since all word vectors are orthogonal and the pairwise euclidean distance between any pair is always $\sqrt{2}$. One-hot vectors are also sparse and inefficient as each newly added word is assigned a new dimension in isolation *i.e.* the dimension of the space increases with the vocabulary size. The prominent approach to representing words in DNNs is to use "continuous" (real valued) word vectors embedded in $|D_X|$-dimensional space with much lower dimensionality than the vocabulary size *i.e.* $|D_X| << |K|$. This is depicted on the right side of Figure 2.12 where 5 words are embedded inside a 3D space. In contrast to binary valued one-hot vectors, real valued continuous representations also allow words to cluster around meaning centroids.

Several techniques allow structuring continuous word spaces specifically through the *distributional hypothesis* which suggests that "words appearing in similar surrounding contexts carry similar semantics" (Harris, 1954; Firth, 1957). *word2vec* (Mikolov et al., 2013) and *GloVe* (Pennington et al., 2014) learn such spaces by making use of very large corpora readily found on the Internet. These models also provide pre-trained word vectors that can be transferred to other language related tasks, similar to how pre-trained CNNs can be used to represent images (section 2.6.3, p. 27). The main approach in NMT is also to use low-dimensional continuous word vectors but by learning them jointly during the training process instead of reusing pre-trained word vectors. From a computational point of view, this is easily achieved by using an *embedding layer* that performs a lookup into a weight matrix $\boldsymbol{E} \in \mathbb{R}^{|K| \times D_X}$ where each row is a D_X-dimensional word embedding. By making $\boldsymbol{E}$ a parameter of the model, the word vectors receive gradient updates leading to a structured word space optimized towards translation performance.

[6]Figure adapted from Holger Schwenk's slides for his talk entitled *Neural Machine Translation and Universal Multilingual Representations.*

The following extends the partial translation example by an embedding layer:

$$
\begin{aligned}
\mathbf{S} &= \left[\texttt{A},\ \texttt{WOMAN},\ \texttt{IS},\ \texttt{PROGRAMMING},\ \texttt{A},\ \texttt{COMPUTER}\right] \\
\mathbf{X} &= \textsf{EMBEDDING}\left(\mathbf{S}\right) \\
&= \left[\boldsymbol{x}_{\texttt{A}},\ \boldsymbol{x}_{\texttt{WOMAN}},\ \boldsymbol{x}_{\texttt{IS}},\ \boldsymbol{x}_{\texttt{PROGRAMMING}},\ \boldsymbol{x}_{\texttt{A}},\ \boldsymbol{x}_{\texttt{COMPUTER}}\right] \\
\mathbf{H} &= \textsf{RNN}\left(\mathbf{X}, \boldsymbol{h_0}\right) \\
\hat{\boldsymbol{y}} &= \textsf{SOFTMAX}\left(\boldsymbol{V}\ \mathrm{H}_{-1} + \boldsymbol{b}_v\right)
\end{aligned}
$$

Vocabulary Granularity

Given a training set, a vocabulary of unique tokens is first constructed prior to training. The size of a vocabulary can range from hundreds to thousands of tokens depending on the size of the training set and the granularity of the vocabulary. The latter defines how aggressively a sentence is segmented into smaller units, such as characters, subwords or words. Although word-level vocabularies are simple to construct and intuitive at first sight, they have limited coverage avoiding them to achieve *open-vocabulary* translation:

- Word-level models can not synthesize novel words: although the model can learn to infer when to output a plural noun based on contextual evidence, they can not achieve this if the plural noun is not available in the vocabulary.
- Whenever a *source* word unknown to the vocabulary is encountered at translation time, the model has no way to represent it in the learned word vector space. Although we reserve a special *out-of-vocabulary* (OOV) embedding, this embedding is never learned during training since every word is "known".

To overcome the coverage problem, subword level segmentation methods (Sennrich et al., 2016; Kudo and Richardson, 2018) are often preferred over word-level vocabularies. Sennrich et al. (2016) proposes an algorithm based on byte pair encoding (BPE) which segments words in the training set sentences based on their corpus frequency: the more frequent a word is, the less likely it will be segmented into smaller subwords. The threshold here is roughly set by a hyperparameter called the number of merge operations which can typically range from 10,000 to 30,000 depending on the size of the dataset. It should be noted that as the segmentation is purely statistical, these methods do not perform a linguistically motivated morphological segmentation. For example, the word "networks" can be splitted as "net - works" although one would expect it to be "network - s". Subword models can synthesize novel surface forms (which are not necessarily valid words) and can represent unknown words by a combination of known subword units in the vocabulary.

2.6 Convolutional Neural Networks

Nowadays it would be surprising to see FCNNs deployed for computer vision tasks even for the previously given simple digit recognition network. The first reason behind this is the relationship between the input size and the model complexity: each neuron in the first hidden layer has as many weights as the number of pixels in the input. For a reasonable hidden layer size of $h{=}512$, the number of parameters jumps from $\sim$1M to $\sim$140M when going from grayscale digit images of size $28{\times}28{\times}1$ to colored real-life images of size $300{\times}300{\times}3$, showing why fully-connected input layers are prohibitive when working with images of variable size. Another drawback of FCNNs is their inability to model hierarchical nature of visual inputs: images are inherently composed of objects which are themselves made of simpler concepts such as edges and primitive geometric patterns. If we would like to detect whether an image contains a "ball" for example, an ideal model should be *translation invariant i.e.* be able to answer independently from the position of the ball. Tightly connecting the neurons to each input pixel is very unlikely to generalize in this case unless the model is exposed to a multitude of training cases with the ball appearing at all possible positions. Convolutional Neural Networks (CNN), which are today used successfully in the literature to process different modalities including images, audio and written language (LeCun et al., 2015), propose a neat solution to these issues using *convolution* and *pooling* operations. Once we fully understand these notions, our previous digit recognition network can be easily extended to incorporate a CNN at the input layer that replaces the inefficient FC input layer.

2.6.1 Convolutional Layers

Let us denote a 2D input of shape[7] $M{\times}M$ with $\boldsymbol{X}$ and a 2D filter of shape $K{\times}K$ with $\boldsymbol{W}$. The *convolution* of the input $\boldsymbol{X}$ with the filter $\boldsymbol{K}$, denoted by $\boldsymbol{X} * \boldsymbol{W}$, produces a feature map $\boldsymbol{F}$ of shape $M'{\times}M'$ where each element is defined as follows[8]:

$$\boldsymbol{F}[i,j] = \sum_{h=0}^{K} \sum_{w=0}^{K} \boldsymbol{X}[i+h, j+w] \boldsymbol{W}[h,w]$$

This is illustrated in Figure 2.13 by a small grid (2x2 filter) sliding over a larger grid (3x3 input) to compute four scalar values that fill an output grid (2x2 feature map). Specifically, each output f_k acts similar to the simple neuron (Equation 2.1, p.5) by computing a dot product between its weights and some part of the input grid.

[7] We limit ourselves to square inputs and filters here since we will be working with square images.
[8] The bias terms are omitted for simplicity.

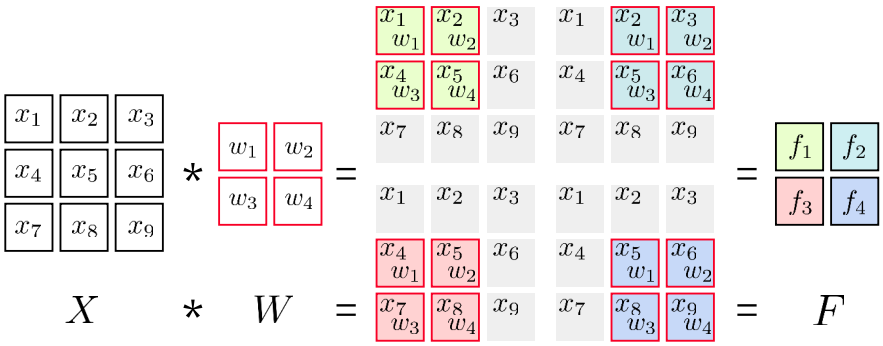


Figure 2.13: The convolution of a 3x3 image with a 2x2 filter yields a 2x2 feature map.

This view allows us to compare the convolution and the simple neuron:

1. The convolution allows *local-connectivity*: the number of parameters in the filter does not have to match the spatial resolution of the input. A valid (albeit larger) feature map is still obtained even the input size is doubled. Often, filters much smaller than the input size are used, mitigating the aforementioned *parameter explosion*.

2. The convolution allows *parameter reuse*: although each output is connected to a different input region, a single set of weights $\{w_i\}$ is shared across the dot products. On the other hand, the neurons in a FC layer do not share parameters.

3. If we set the filter size equal to the input size, a single dot product f_1 comes out of the convolution hence f_1 becomes a fully-connected neuron.

A 2D *convolutional layer* is a computational unit composed of at least one filter, where filters extend towards a third dimension which is the channel dimension C. An input and a filter are now respectively denoted by $\mathbf{X} \in \mathbb{R}^{M \times M \times C}$ and $\mathbf{W}_i \in \mathbb{R}^{K \times K \times C}$ where channel dimensions for both should match. The convolution of $\mathbf{X}$ with a filter $\mathbf{W}_i$ yields a 2D feature map $\boldsymbol{F}_i \in \mathbb{R}^{M' \times M'}$ where each element is the dot product between the filter weights and the corresponding input volume. A layer with C' filters then produces C' feature maps $\{\boldsymbol{F}_i\}_{i=1}^{C'}$ which when stacked together, forms an output volume $\mathbf{F} \in \mathbb{R}^{M' \times M' \times C'}$. Figure 2.14 illustrates a convolutional layer where a 6x6x3 image input is transformed with C'=4 different filters of size 3x3x3 each. A convolutional layer is often followed by a non-linearity such as ReLU (He et al., 2015). This combination intrinsically behaves like a visual pattern detector which fires to specific patterns highlighted by the convolution.

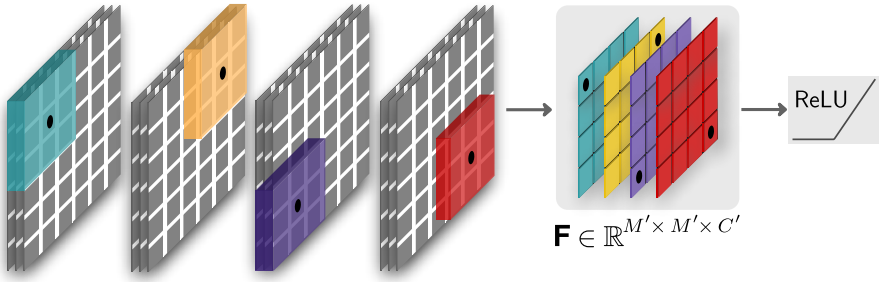


Figure 2.14: A ReLU convolutional layer with 4 filters of size 3x3x3: each (colored) filter applies a convolution with outputs indicated (•) in the corresponding feature maps.

2.6.2 Pooling Layers

One side-effect of the previously described convolution operation is how it shrinks the spatial resolution of the input from $M \times M$ to $M' \times M'$ where $M' < M$. This is generally an unwanted effect that hinders the design of deep architectures since stacking more convolutional layers will quickly shrink the input image to 1x1. The common practice is then to pad the input to the convolution layers with explicit zero pixels so that $M'=M$ and delegate the shrinking to *pooling* layers when required (Goodfellow et al., 2016). These special layers independently operate on top of each feature map to summarize/fuse the local activation neighborhoods. Specifically, they again operate over small regions like the convolution except that the filter is no longer learned through backpropagation. For example, an average pooling of size 3×3 will convolve a filter pre-filled with $\frac{1}{9}=\frac{1}{3\times 3}$ to compute an average activation over the region. Modern CNNs generally use *max-pooling* which instead of taking an average, selects the highest activation as the region summary. The *translation invariance* property of CNNs (see the "ball" example in section 2.6) is often attributed to *max-pooling* since a shift to the most activated neuron in the input region does not influence to output of the pooling. A variant of *average pooling* called *global average pooling* (GAP) is often used after the last convolutional layer in the network in order to produce a global vector that will be further projected to the number of classes defined for the task.

In summary, deep CNNs are able to learn a hierarchical decision function where the deeper layers detect complex patterns which are themselves composed of simpler patterns (Figure 2.15). This is supported by early studies in neuroscience as well: Hubel and Wiesel (1962) discovered that the visual cortex of the cat contains simple and complex cells which respond to visual stimuli in increasing levels of complexity ranging from light intensity changes to geometric patterns. In fact, as one of the very first pattern detection

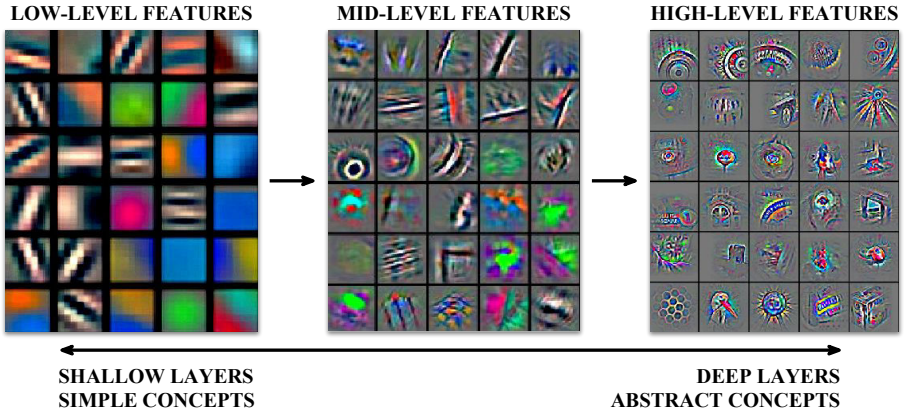


Figure 2.15: The feature compositionality of deep CNN models: high-level abstract concepts are represented using simpler ones. Figure adapted from Zeiler and Fergus (2014).

networks in the literature, *neocognitron* (Fukushima, 1980) already integrated locally-connected units and pooling layers bearing a strong resemblance to modern CNNs except that it lacked backpropagation.

2.6.3 Pre-trained CNNs as Feature Extractors

Being able to categorize images of real-life objects is a relatively hard task to solve for an AI system. The factors affecting its difficulty range from the level of detail and complexity present in the images, to the size of the visual vocabulary *i.e.* the number of possible labels that can be assigned. An influential resource in this respect is the ImageNet dataset (Deng et al., 2009) which comprises 1.2 millions training images hand-labeled with 1000 object categories. Together with the periodically held "ImageNet Large Scale Visual Recognition Challenge" (ILSVRC) (Russakovsky et al., 2015), this dataset fostered research in computer vision especially in the context of image classification and object localization. For the first time in 2012, a deep CNN architecture called AlexNet (Krizhevsky et al., 2012) won the competition by increasing the Top-5 classification accuracy around 11% compared to previous non-neural approaches. The following 5 years of ILSVRC witnessed an unprecedented progress in classification performance thanks to deeper and more parameter efficient CNN architectures such as 19-layers VGGNet (Simonyan and Zisserman, 2014), 152-layers ResNet (He et al., 2016), and DenseNet which goes beyond 200-layers (Huang et al., 2017b). The Top-5 classification accuracy achieved by these models are in the range of 92-96%. The 34-layers variant of the ResNet is depicted in Figure 2.16. Following the success of deep CNNs in large scale image classification, there has been growing interest in reusing their intermediate representations in other AI tasks. Here we should make a distinction between two such intermediate CNN representations:

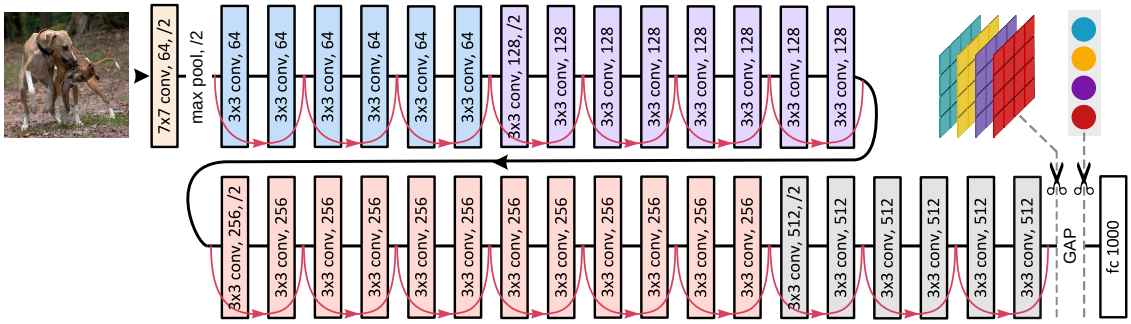


Figure 2.16: 34-layer ResNet CNN with residual connections (He et al., 2016). Once trained, features are generally extracted from before or after the final GAP layer.

- **Spatial features** $\left(\mathbf{F} \in \mathbb{R}^{M' \times M' \times C'}\right)$ are feature maps extracted from an arbitrary convolutional layer. Since achieving high accuracy on ILSVRC would necessite a global visual understanding, spatial features are believed to be rich enough to help auxiliary tasks. For instance, Xu et al. (2015b) used spatial features for the first time to do image captioning by glimpsing over regions in the image *i.e.* a mechanism called *visual attention*. It is a common practice to target "late" convolutional layers for extraction (Figure 2.16) in order to obtain high-level/conceptual features.

- **Global features** $\left(\boldsymbol{f} \in \mathbb{R}^{C'}\right)$ on the other hand are more abstractive and optimized towards the original task since they are extracted from before the output layer. In the case of ResNet, these features are nothing more than global average poolings of final convolutional feature maps (Figure 2.16). Despite their simplicity, Razavian et al. (2014) showed how a linear classifier on top of them results in superior performance compared to previous state-of-the-art in tasks such as scene classification and image retrieval. Early works in image captioning successfully made use of these features as well (Kiros et al., 2014; Mao et al., 2015; Vinyals et al., 2015).

With all the evidence hinting at the expressiveness of pre-trained visual features, we will be experimenting with both spatial and global features for multimodal translation.

2.7 Summary

In this chapter, I first described the building blocks necessary to construct fully connected DNNs in supervised learning framework and the notions of objective function and stochastic parameter optimization. After giving a complete recipe that uses backpropagation and the SGD algorithm to train a DNN, I proceeded with the detailed explanations of RNNs and CNNs that will be extensively used to represent the visual modality and linguistic inputs such as sentences. Based on the background provided, the following chapter will introduce the current state-of-the-art in neural machine translation.

CHAPTER 3

Neural Machine Translation

A machine translation (MT) system is a computer system that automatically translates content from one language to another without any human intervention. Inspired by the previous successes in cryptography, American scientist Warren Weaver claimed about the possibility of such systems for the first time in 1947: "*When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'*" Today, MT systems are capable of producing decent translations that may require minimum to none post-editing effort, thanks to the massive amounts of publicly available bilingual data and powerful software and hardware components.

Two radically different approaches currently dominate the field: phrase-based machine translation (PBMT) (Koehn et al., 2003) and neural machine translation (NMT) based on DNNs (Cho et al., 2014b; Bahdanau et al., 2014; Sutskever et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). Although PBMTs seem to have their own advantages over NMTs in low-resource conditions or in terms of out-of-domain translation performance (Koehn and Knowles, 2017), NMT is now considered the prominent approach in the field both actively researched and also deployed in many online translation services such as *Google Translate* and *Microsoft Translator*. NMT is an encoding & decoding machinery largely compatible with what Weaver previously suggested: an encoder encodes a sentence into an intermediate latent representation which is further consumed by the decoder to generate an appropriate translation. We can draw parallels between this intermediate representation and the concept of interlingual MT that encodes to and decodes from a language-agnostic meaning representation called an *interlingua* (Delavenay and Delavenay, 1960). Although the difficulty of manually constructing a rule-based interlingua limited these early systems to simplistic ad-hoc translation problems (Nyberg and Mitamura, 1992), the idea itself remains elegant and is more likely to be exploitable by the differentiable nature of NMTs forcing the model to obtain useful translation-oriented

latent representations. In fact, multilingual NMTs (Ha et al., 2016; Johnson et al., 2016; Firat et al., 2017) demonstrated that such universal representations can indeed be learned by a single NMT system when trained on a combination of input and output languages. The end-to-end & differentiable nature of neural systems provides an exceptional flexibility for exploring novel architectures integrating multiple modalities as well. This thesis is not an exception to the ongoing neural trend as the MMT systems that we will be exploring in the next chapters are pure extensions to the existing NMT systems.

In this chapter, I will first start by introducing neural language modeling (NLM) which provides a generalized framework for formulating NMTs as *conditional* language models. After briefly describing PBMTs, I will focus on the basic sequence-to-sequence NMT architecture (Cho et al., 2014b; Sutskever et al., 2014) followed by its attentive extension (Bahdanau et al., 2014). Lastly, I will talk about how to decode translations from an NMT system and introduce automatic evaluation metrics commonly used for MT evaluation. In the following, $\mathbf{X} = [\mathrm{X}_1, \ldots, \mathrm{X}_S]$ and $\mathbf{Y} = [\mathrm{Y}_1, \ldots, \mathrm{Y}_T]$ denote the source and target sequences where each individual token X_s and Y_t belong to the source and target vocabularies $\mathbb{S}$ and $\mathbb{T}$, respectively.

3.1 The Language Modeling Perspective

The purpose of a language model (LM) is to estimate the probability of a sequence where the definition of a sequence can range from sentences to large documents. Once trained, an LM can be used to predict the next token given the previous ones or can answer the question of "how likely is it to encounter this sequence?" by assigning a score to it. Formally, the sequence probability $P(\mathbf{Y})$ can be decomposed into T conditional probabilities where each term is conditioned on the full previous context denoted by $\mathrm{Y}_{<t}$:

$$
\begin{aligned}
P(\mathbf{Y}) &= \prod_{t=1}^{T} P(\mathrm{Y}_t|\mathrm{Y}_{<t}) = \prod_{t=1}^{T} P(\mathrm{Y}_t|\mathrm{Y}_1, \ldots, \mathrm{Y}_{t-1}) \\
&= P(\mathrm{Y}_1)P(\mathrm{Y}_2|\mathrm{Y}_1)P(\mathrm{Y}_3|\mathrm{Y}_1, \mathrm{Y}_2)\ldots P(\mathrm{Y}_t|\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_{t-1})
\end{aligned}
\tag{3.1}
$$

Traditional n-gram LMs approximate this probability by relaxing the full context to a fixed-size context of n previous tokens where n is typically three or four:

$$
P(\mathbf{Y}) = \prod_{t=1}^{T} P(\mathrm{Y}_t|\mathrm{Y}_{<t}) \approx \prod_{t=1}^{T} P\left(\mathrm{Y}_t \,|\, \mathrm{Y}_{t-1}, \ldots, \mathrm{Y}_{t-n+1}\right)
$$

n-gram LMs are powerful non-parametric models purely estimated from count statistics of a given monolingual corpus. Although powerful and widely used, they lack the potential benefits of full context and requires the integration of techniques such as backing-off and smoothing (Kneser and Ney, 1995) to prevent the *sparsity* problem *i.e.* the underestimation of rare or never occurring n-grams. DNN-based LMs (NNLM) attempted to resolve this sparsity problem by representing the context with the concatenation of word embeddings associated to n previous tokens (Bengio et al., 2003; Schwenk, 2010). This context vector is then non-linearly transformed and projected to the size of the vocabulary for further probability estimation using the same multi-class classification framework introduced in the previous chapter. This way, an n-gram that never occurred in the training set can still be represented if its constituents are known to the model. However, as the number of parameters in the non-linear layer depends on the size of the context n, these NNLMs were practically limited to fixed-size contexts as well. In 2010, Mikolov et al. proposed a recurrent LM (RNNLM) that encodes variable-length sequences, making it possible to use arbitrarily long contexts instead of a predetermined context size. An RNNLM estimates the probability of a single sequence $\mathbf{Y}$ as follows:

$$
\begin{aligned}
\mathbf{Y} &= [\text{A, WOMAN, PLAYS, TENNIS, <eos>}] && \text{OUTPUT} \\
\mathbf{Y}' &= [\text{<bos>, A, WOMAN, PLAYS, TENNIS}] && \text{INPUT} \\
[\boldsymbol{h_1}, \dots, \boldsymbol{h_T}] &= \mathsf{RNN}\left(\mathsf{EMB}\left(\mathbf{Y}'\right), \boldsymbol{h_0}\right) && \text{ENCODE} \\
\boldsymbol{c_t} &= f(\boldsymbol{h_t}) && \text{CONTEXT } \textit{i.e.}\ \mathrm{Y}_{<t} \qquad (3.2) \\
P(\mathbf{Y}) &= \prod_{t=1}^{T} P\left(\mathrm{Y}_t \mid \mathrm{Y}_{<t}\right) = \prod_{t=1}^{T} P\left(\mathrm{Y}_t \mid \boldsymbol{c_t}\right) && \text{SEQUENCE PROB.} \qquad (3.3) \\
-\log(P(\mathbf{Y})) &= -\sum_{t=1}^{T} \log\left(P(\mathrm{Y}_t \mid \boldsymbol{c_t})\right) && \text{SEQUENCE NLL}
\end{aligned}
$$

Note how we were able to define a pseudo-input sequence $\mathbf{Y}'$ which is actually a time-shifted version of $\mathbf{Y}$. This allows us to formulate the RNNLM as a mapping from an input sequence to an output sequence, consistent with the notational framework introduced in the first chapter. Special tokens such as <bos> and <eos> are generally used to mark the "beginning" and the "end" of the sequences. Each new hidden state produced by the RNN conveys information about the sequence processed so far *i.e.* the full-context $\mathrm{Y}_{<t}$. This is why we represent the context as a function of the recurrent hidden state $\boldsymbol{h_t}$ (equation 3.2) where f is an arbitrarily complex output block that projects $\boldsymbol{h_t}$ to the size of the vocabulary. At each timestep $P(\mathrm{Y}_t \mid \boldsymbol{c_t})$ estimates the probability that corresponds to the true token Y_t by applying softmax normalization. We finally obtain the training set NLL by simply averaging the sequence NLLs.

3.2 Phrase-based MT

PBMTs formulate the translation problem as a probability distribution which is now conditioned on the source side *i.e.* $P(\mathbf{Y}|\mathbf{X})$ (Koehn et al., 2003). This conditional probability is further factorized into a "translation model (TM)" $P(\mathbf{X}|\mathbf{Y})$ and a "target LM" $P_{\text{LM}}(\mathbf{Y})$ component. In practice however, the factorization is often expressed as a weighted log-linear model with weights λ_i assigned to feature functions f_i:

$$P(\mathbf{Y}|\mathbf{X}) = P(\mathbf{X}|\mathbf{Y})\, P_{\text{LM}}(\mathbf{Y})$$
$$\log\left(P(\mathbf{Y}|\mathbf{X})\right) = \sum_{i=1}^{N} \lambda_i f_i(\mathbf{X}, \mathbf{Y}) + \lambda_{\text{LM}} f_{\text{LM}}(\mathbf{Y})$$

A feature function is a subcomponent that scores a given source-candidate pair with respect to a specific aspect of the translation problem such as how well the words are aligned to each other. The TM component is essentially a feature function as well that estimates the likelihood of a target phrase given a source one, using the phrase table it constructs from the parallel training corpora. The LM component on the other hand is generally learned on a large, separate monolingual corpus in the target language so that it can be used to score the translation candidates with respect to their fluency. The weights λ_i are optimized (Och, 2003) to maximize the translation quality on a held-out development set using evaluation metrics such as BLEU (Papineni et al., 2002). Finally, the best translation $\mathbf{Y}^*$ is obtained by searching through the hypothesis space of the model to satisfy the following:

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} \; \log\left(P(\mathbf{Y}|\mathbf{X})\right) \tag{3.4}$$

PBMTs are complex systems that incorporate many feature functions carefully designed by experts throughout years of research (Koehn, 2010). NMTs instead, propose to replace the whole pipeline used to estimate $P(\mathbf{Y}|\mathbf{X})$ with an end-to-end DNN that implicitly replaces the LM component as well.

3.3 Early Neural Approaches

Prior to purely end-to-end NMT models, there has been several attempts to couple DNNs and traditional MT systems. Schwenk (2012) proposed a DNN similar to NNLM (Bengio et al., 2003) that estimates the phrase translation probabilities of a PBMT: the model projects all words in a source phrase into a continuous vector from which a joint distribution of words in the target phrase is estimated. The author provided empirical evidence

that the system was able to provide meaningful phrase translations even for unseen source phrases.

Another type of coupling exploited the distributional power of DNNs to rescore an n-best list *i.e.* a set of candidate translations obtained from a traditional MT system. The top candidate translation after reordering is considered as the final translation. The neural network joint model (NNJM) (Devlin et al., 2014) is one such method that extends the NNLM by augmenting the n-gram target context with an m-gram source context $\overline{\mathbf{X}}_t$:

$$P(\mathbf{Y}|\mathbf{X}) \approx \prod_{t=1}^{T} P\left(\mathrm{Y}_t \,|\, \mathrm{Y}_{t-1}, \ldots, \mathrm{Y}_{t-n+1}, \overline{\mathbf{X}}_t\right) \tag{3.5}$$

The model uses external word alignments in order to select the best possible source context window $\overline{\mathbf{X}}_t$ at each timestep t. The final context vector is exactly formed as in NNLM *i.e.* by concatenating all embeddings related to the source and target contexts. The authors showed significant improvements over a state-of-the-art MT in Arabic-English and Chinese-English translation tasks.

3.4 Sequence-to-Sequence NMT

We define a sequence-to-sequence (S2S) NMT any neural system that reads a source sequence and then translates it into a target sequence. The first attempt to S2S NMT came from Kalchbrenner and Blunsom (2013) where the authors proposed two different models which both utilize a CNN encoder and an RNNLM decoder. Unlike the n-gram relaxation in equation 3.5, the RNNLM decoder here has access to full-context $\mathrm{Y}_{<t}$. The first model encodes a source sentence with a CNN to obtain a constant source context vector from which the RNNLM decodes the translation whereas the second one replaces the constant context with a dynamic n-gram one represented as convolutional feature maps. The results mostly focused on rescoring performance and on the sensitivity of the models to the source word order. Later on, Cho et al. (2014a,b) proposed a very similar S2S architecture to Kalchbrenner and Blunsom (2013) by replacing the CNN and the RNNLM components with their novel GRU layer (section 2.5.1, p. 20). This model along with the concurrent work by Sutskever et al. (2014) are considered the first successful encoder-decoder NMTs in the literature. I will now proceed with a detailed description of encoder-decoder NMTs since the multimodal architectures designed throughout this thesis are derivations of them.

3.4.1 Recurrent Encoder

A recurrent encoder encodes a given sequence $\mathbf{X}$ to a sequence of hidden states $\mathbf{H}$ by using a recurrent layer $\mathbf{e}()$ such as a plain RNN or a gated variant GRU (Cho et al., 2014b) or LSTM (Sutskever et al., 2014). For simplicity, the source embedding layer is also made part of the encoder so that a sequence $\mathbf{X}$ of one-hot encoded tokens is implicitly mapped to continuous token representations before further processing (section 2.5.2, p. 21). The following example illustrates the sequence of operations performed by the encoder ENC:

$$
\begin{aligned}
\mathbf{X} &= [\text{A, WOMAN, PLAYS, TENNIS, <eos>}] \\
\mathbf{H} &= \mathsf{ENC}\left(\mathbf{X}, \boldsymbol{h_0} \leftarrow 0\right) \\
&= \mathbf{e}\left(\mathsf{EMB}\left(\mathbf{X}\right), \boldsymbol{h_0} \leftarrow 0\right) \\
&= \mathbf{e}\left([\boldsymbol{x_1}, \ldots, \boldsymbol{x_S}], \boldsymbol{h_0} \leftarrow 0\right) \\
\mathbf{H} &= [\boldsymbol{h_1}, \ldots, \boldsymbol{h_S}]
\end{aligned}
$$

The end of source sequences is explicitly tagged with an <eos> token so that the encoder can learn how sentences come to an end, which probably is useful in estimating the target sentence length during translation generation. The initial hidden state $\boldsymbol{h_0}$ is often set to 0 unless otherwise stated. Each produced encoding $\boldsymbol{h_i}$ conveys information about the phrase processed so far up to that position including the token X_i itself. We assume that an encoder always provides the full set of encodings $\mathbf{H}$ and we delegate the choice of source context type to the decoder.

Bidirectional Encoding

The RNNs process an input sequence from left-to-right in a unidirectional fashion. This means that the last hidden state is fully aware of the past context while the earlier ones have more and more limited context. In the limit, the first hidden state $\boldsymbol{h_1}$ has no access to any contextual information making it a mere word encoding. Bidirectional RNNs (Schuster and Paliwal, 1997) propose a simple extension to unidirectional RNNs by sparing a dedicated RNN for right-to-left encoding. At a given timestep, the encoding $\boldsymbol{h_i}$ now doubles its size by concatenating the left-to-right and right-to-left hidden states obtained from these two RNNs. By denoting the original and the reversed sequence with $\overrightarrow{\mathbf{X}}$ and $\overleftarrow{\mathbf{X}}$ respectively, the encodings produced by a bidirectional GRU encoder are given as follows:

$$
\mathbf{H} = \left[\begin{pmatrix} \overrightarrow{\boldsymbol{h_1}} \\ \overleftarrow{\boldsymbol{h_1}} \end{pmatrix}, \ldots, \begin{pmatrix} \overrightarrow{\boldsymbol{h_S}} \\ \overleftarrow{\boldsymbol{h_S}} \end{pmatrix} \right] = \begin{bmatrix} \overrightarrow{\mathsf{GRU}}\left(\overrightarrow{\mathbf{X}}, \overrightarrow{\boldsymbol{h_0}}\right) \\ \overleftarrow{\mathsf{GRU}}\left(\overleftarrow{\mathbf{X}}, \overleftarrow{\boldsymbol{h_0}}\right) \end{bmatrix}
$$

From now on, all recurrent encoders are assumed to be bidirectional although we do not explicitly precise this in the equations and the figures for the sake of simplicity.

3.4.2 Recurrent Decoder

A recurrent decoder generates the target sequence one token at a time given the previous tokens $\mathrm{Y}_{<t}$ (equivalent to $\mathrm{Y}'_{\leq t}$) and a representation of the source sentence. Both Cho et al. (2014b) and Sutskever et al. (2014) propose to compress the whole source sentence into a *constant* context *i.e.* a single high dimensional vector $\boldsymbol{c}$ that does not evolve across decoding timesteps. Specifically, Cho et al. (2014b) defines the source context as a function of the last encoding ($\mathrm{H}_{-1} = \boldsymbol{h}_{\mathbf{S}}$) as follows:

$$\boldsymbol{c} = \tanh\left(\boldsymbol{W}_c \, \mathrm{H}_{-1} + \boldsymbol{b}_c\right)$$

Besides conditioning the decoder through its initial hidden state $\boldsymbol{h}'_{\mathbf{0}}$, Cho et al. (2014b) also redefines the GRU logic in the decoder so that $\boldsymbol{c}$ is concatenated to the hidden state at each timestep t. This ensures that the impact of $\boldsymbol{c}$ does not vanish across the recurrence in the decoder.

The following illustrates the sequence of operations performed by the decoder DEC. The input to the decoder is a time-shifted version $\mathbf{Y}'$ of the true target sequence $\mathbf{Y}$. $\mathbf{Y}'$ begins with the <bos> token to explicitly trigger a sentence start. The recurrent layer $\mathbf{d}()$ represented here is can be again any RNN variant:

$$\begin{aligned}
\mathbf{Y} &= [\text{UNE, FEMME, JOUE, AU, TENNIS, <eos>}] \\
\mathbf{Y}' &= [\text{<bos>, UNE, FEMME, JOUE, AU, TENNIS}] \\
\mathbf{H}' &= \mathsf{DEC}\left(\mathbf{Y}', \boldsymbol{h}'_{\mathbf{0}} \leftarrow \boldsymbol{c}\right) \\
&= \mathbf{d}\left(\mathsf{EMB}\left(\mathbf{Y}'\right), \boldsymbol{h}'_{\mathbf{0}} \leftarrow \boldsymbol{c}\right) \\
\mathbf{H}' &= [\boldsymbol{h}'_{\mathbf{1}}, \ldots, \boldsymbol{h}'_{\mathbf{T}}]
\end{aligned} \tag{3.6}$$

Note how the model is consistently trained with the embeddings of the true previous tokens (equation 3.6), a technique called *teacher-forcing* (Goodfellow et al., 2016). When decoding translations however, the model has to receive its previous predictions since the true distribution is unknown. It has been shown that gradually exposing the model to its own mistakes – a technique called *scheduled sampling* – alleviates this problem and improves the performance (Bengio et al., 2015).

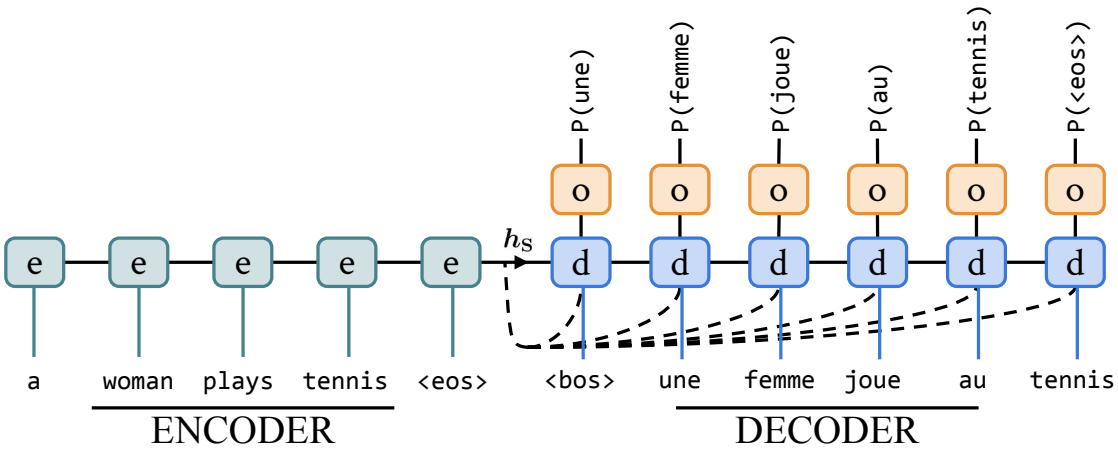


Figure 3.1: NMT with constant source context: the (**e**)ncoder and the (**d**)ecoder are unfolded along the time axis. The dashed connections show the additional context inputs to GRU (Cho et al., 2014b). The orange **o** layer is the output logic.

Output Logic

Once the hidden states $\mathbf{H}'$ are obtained from the decoder, an output layer is used to project them into the size of the target vocabulary. This can be realized with a simple FC layer or a complex one such as the *deep output* (Pascanu et al., 2014) used by Cho et al. (2014b). Although the hidden state $\boldsymbol{h}'_{\mathbf{t}}$ is already conditioned on the source context and the previous embedding intrinsically, deep output creates a residual link to the encoder and to the target embedding layer to alleviate possible vanishing gradients:

$$\begin{aligned} \boldsymbol{o}_{\mathbf{t}} &= \tanh\left(\boldsymbol{V}_h \boldsymbol{h}'_{\mathbf{t}} + \boldsymbol{V}_y \boldsymbol{y}_{\mathbf{t-1}} + \boldsymbol{V}_c \boldsymbol{c}\right) \\ P\left(\mathrm{Y}_t \,|\, \mathrm{Y}_{<t}, \mathbf{X}\right) &= \textsc{softmax}\left(\boldsymbol{W}_o \boldsymbol{o}_{\mathbf{t}}\right) \end{aligned} \tag{3.7}$$

The final linear transformation $\boldsymbol{W}_o$ which projects the output $\boldsymbol{o}_{\mathbf{t}}$ to the size of the target vocabulary is generally considered a secondary embedding matrix referred to as *output embeddings*. If the size of the output vector $\boldsymbol{o}_{\mathbf{t}}$ is set to be equal to the size of a target word embedding, the two embedding layers in the decoder can be shared so that a single embedding matrix is learned for both purposes. This is called *tied embeddings* (Inan et al., 2016; Press and Wolf, 2016) and shrinks down the number of parameters in an NMT substantially if the size of the vocabulary is very large. Figure 3.1 shows the complete computation graph from the encoder to the probability distribution $P\left(\mathrm{Y}_t \,|\, \mathrm{Y}_{<t}, \mathbf{X}\right)$. The training NLL is then computed as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \log(P(\mathbf{Y}^{(i)})) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \log\left(P(\mathrm{Y}_t^{(i)} \,|\, \mathrm{Y}_{<t}^{(i)}, \mathbf{X}^{(i)})\right)$$

3.4.3 Attention Mechanism

The NMT model described so far is limited in a way that it encodes arbitrarily long sentences into a single vector $\boldsymbol{c}$. This bottleneck makes it hard for the model to come up with an encoding scheme that can encode both a very short and a very long sequence in an equally expressive way. In fact, Cho et al. (2014a) showed how the performance of the encoder-decoder NMT sharply decreases as the sentence length increases, unlike PBMT systems that are almost invariant to the sentence length. The attention mechanism (Bahdanau et al., 2014) provides a nice solution to the problem by replacing the single vector $\boldsymbol{c}$ with a dynamic and time-dependent one $\boldsymbol{c_t}$. This allows the decoder to look at different portions of the source sentence as the decoding progresses. The authors show that the addition of the attention mechanism combined with a bidirectional encoder mitigates the performance collapse that occurs as the sentences get longer. Today, state-of-the-art NMTs are equipped with attention mechanisms between the encoder and the decoder and even in other components of the network (section 3.4.6, p. 40).

Formally, at each timestep t of the decoding process, the attention mechanism receives the hidden state $\boldsymbol{h'_t}$ of the decoder as a "query" vector and computes a relevance score between each encoding $\boldsymbol{h_i} \in \mathbf{H}$ and the query. The time-dependent context $\boldsymbol{c_t}$ is then computed as the weighted sum of encodings where the weights are the normalized relevance scores that sum to one:

$$
\begin{aligned}
z_i &= \textsc{score}\left(\boldsymbol{h_i}, \boldsymbol{h'_t}\right) \\
\boldsymbol{\alpha} &= [\alpha_1, \ldots, \alpha_S]^\top = \textsc{softmax}\left([z_1, \ldots, z_S]^\top\right) \\
\boldsymbol{c_t} &= \mathbf{H}\boldsymbol{\alpha} = \sum_i^S \alpha_i \, \boldsymbol{h_i}
\end{aligned}
\tag{3.8}
$$

Two common methods exist for computing the relevance scores: Bahdanau et al. (2014) propose a parameterized FF layer while Luong et al. (2015b) simply use the dot product (Figure 3.2). In the context of NMT, both methods have been shown to perform equally well (Britz et al., 2017). The following illustrates both approaches at decoding timestep t using a query vector $\boldsymbol{h'_t}$ and a single encoding $\boldsymbol{h_i}$. The linear transformations $\boldsymbol{W}_e$ and $\boldsymbol{W}_q$ are used to project the encoding and the query to a common space:

$$
\begin{aligned}
z_i &= \textsc{score}\left(\boldsymbol{h_i}, \boldsymbol{h'_t}\right) \\
&\rightarrow \boldsymbol{w}_a^\top \tanh(\boldsymbol{W}_e \, \boldsymbol{h_i} + \boldsymbol{W}_q \, \boldsymbol{h'_t}) && \text{Bahdanau et al. (2014)} \\
&\rightarrow (\boldsymbol{W}_e \, \boldsymbol{h_i})^\top (\boldsymbol{W}_q \, \boldsymbol{h'_t}) && \text{Luong et al. (2015b)}
\end{aligned}
$$

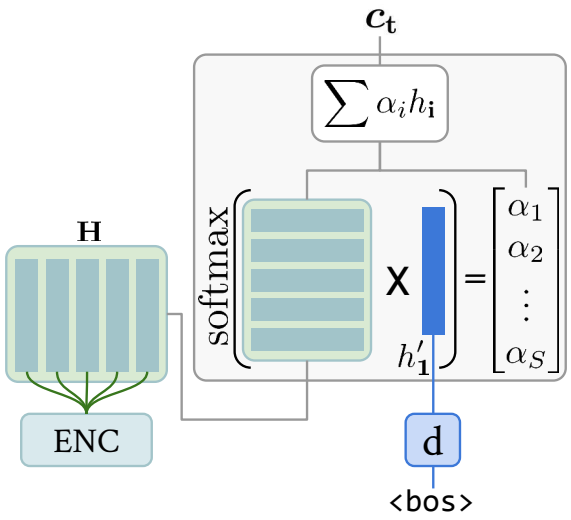


Figure 3.2: A decoding timestep ($t = 1$) with "dot-attention" (Luong et al., 2015b): The transformations to the query and the encodings are omitted for simplicity.

We can now wrap all the underlying attention computations into a layer function $\mathbf{a}()$ and modify the equation 3.7 to additively integrate the context into the output logic:

$$\boldsymbol{c_t} = \mathbf{a}\left(\mathbf{H}, \boldsymbol{h'_t}\right)$$
$$\boldsymbol{o_t} = \tanh\left(\boldsymbol{O}_h\, \boldsymbol{h'_t} + \boldsymbol{O}_y\, \boldsymbol{y_{t-1}} + \boldsymbol{O}_c\, \boldsymbol{c_t}\right)$$

In models with more than one recurrent layers, the context $\boldsymbol{c_t}$ is often propagated to the subsequent layers as the input. One such example is the Conditional GRU model that will be explained in the next section.

3.4.4 Conditional GRU Decoder

The conditional GRU (CGRU) implements a slightly different decoder logic with two GRU layers encapsulating the attention mechanism (Sennrich et al., 2017). The recurrent hidden states of the GRUs are "transitional" in the sense that the previous hidden state of the second GRU is determined by the first GRU. The second GRU then computes the new hidden state that becomes the previous hidden state of the first GRU in the next timestep (Figure 3.3). The input to the second GRU is the context $\boldsymbol{c_t}$ computed by the attention layer.

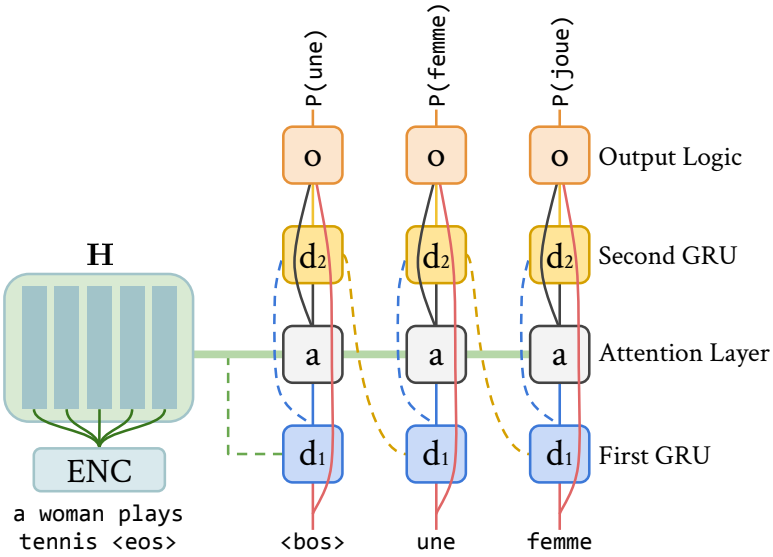


Figure 3.3: Conditional GRU decoder: The hidden state of the first GRU ($\mathbf{d_1}$) becomes the query for the attention. The context $\boldsymbol{c_t}$ produced by the attention is fed to the second GRU ($\mathbf{d_2}$) as the input. The dashed connections refer to hidden state transitions.

The following summarizes the CGRU NMT for the initial decoding timestep $t = 1$:

$$
\begin{aligned}
\mathbf{H} &= \text{ENC}\left(\mathbf{X}, \boldsymbol{h_0} \leftarrow 0\right) \\
\boldsymbol{h'_0} &= \tanh\left(\boldsymbol{W}_c \left\{\mathrm{H}_{-1} \text{ or } \mathrm{H}_{\text{AVG}}\right\} + \boldsymbol{b}_c\right) && (3.9) \\
\boldsymbol{h'_1} &= \mathbf{d_1}\left(\text{EMB}\left(\text{<bos>}\right), \boldsymbol{h'_0}\right) && (3.10) \\
\boldsymbol{c_1} &= \mathbf{a}\left(\mathbf{H}, \boldsymbol{h'_1}\right) \\
\boldsymbol{h''_1} &= \mathbf{d_2}\left(\boldsymbol{c_1}, \boldsymbol{h'_1}\right) \\
\boldsymbol{o_1} &= \tanh\left(\boldsymbol{O}_h\, \boldsymbol{h''_1} + \boldsymbol{O}_y\, \text{EMB}\left(\text{<bos>}\right) + \boldsymbol{O}_c\, \boldsymbol{c_1}\right) \\
P\left(\mathrm{Y}_1 \,|\, \text{<bos>}, \mathbf{X}\right) &= \text{SOFTMAX}\left(\boldsymbol{W}_o\, \boldsymbol{o_1}\right)
\end{aligned}
$$

$\boldsymbol{h'}$ and $\boldsymbol{h''}$ denote the hidden states of $\mathbf{d_1}$ and $\mathbf{d_2}$, respectively. Common choices when setting $\boldsymbol{h'_0}$ in equation 3.9 are the last (H_{-1}) or the average (H_{AVG}) encoding although with attention, we observe little to none performance drop even it is set to $\mathbf{0}$.

3.4.5 Deep Models

The depth of an NMT model can be quantified by how many layers are used to process the source and target sequences. The main model described (Cho et al., 2014b) is a shallow NMT as both the encoder and the decoder consist of a single GRU layer. On the other hand, the model proposed by Sutskever et al. (2014) is a deep NMT as the encoder

and the decoder are each constructed by stacking four LSTM layers. When many recurrent layers are stacked this way, each layer receives as input the set of encodings $\mathbf{H}$ produced by the previous layer except the first layer which receives the input sequence $\mathbf{X}$. The depth especially becomes an important factor for large-scale state-of-the-art deployments (Johnson et al., 2016; Gehring et al., 2017; Vaswani et al., 2017).

3.4.6 Non-recurrent Approaches

The sequential nature of RNNs prevents them from being parallelized across multiple devices during training. The parallelization is especially important when training large-scale deep NMTs on massive amounts of parallel data, often in the order of millions of sentences. There has been many attempts to replace RNNs with deep CNNs and FCNNs: Gehring et al. (2017) replace them by convolutional layers while Vaswani et al. (2017) employ very deep FCNN encoders and decoders with a variant of attention called "self-attention". When applied on top of a set of hidden states, "self-attention" computes the relevance of each one of them to the set of hidden states themselves. Since MT is a translation-variant problem, the lack of recurrent processing is often remedied by explicitly encoding word positions through the use of special "positional embeddings".

3.4.7 Multitask Learning for NMT

Multitask Learning (MTL) (Caruana, 1997) is a learning paradigm where related tasks are trained in a parallel fashion. An MTL architecture generally passes through a common representation which is shared across the tasks and which encodes domain/modality relevant knowledge useful to improve final generalization performance. Dong et al. (2015) successfully used MTL to learn a one-to-many NMT with a shared recurrent encoder and multiple target language decoders with dedicated attention mechanisms. At training time, they form minibatches containing sentences from one language pair only and this language pair is randomly sampled at each iteration. In the end, the parameters of the shared encoder are always updated during the backward-pass while the decoders are selectively updated depending on the language pair considered. Luong et al. (2015a) further extended Dong et al. (2015) to many-to-one and many-to-many setups with tasks ranging from translation to captioning and parsing.

3.5 Evaluation of MT Outputs

Classical machine learning metrics such as precision, accuracy or recall are not directly applicable to sequence transduction problems where the output is a sequence of tokens. Although the gold standard for MT evaluation is manual evaluation, we need a cheap and easy way to approximately assess the quality of the obtained translations in order to evaluate, compare and select MT models. This is achieved by automatic metrics that measure the similarity between the machine generated translations and the reference sentences translated by human annotators. The most commonly used automatic metric in MT is BLEU (Papineni et al., 2002) which is a document-level metric that computes the geometric mean of n-gram matching precisions (up to 4-gram precision in the default setting) between the reference sentences and the MT outputs. Another metric often used for evaluating image captioning and multimodal machine translation systems is METEOR (Lavie and Agarwal, 2007; Denkowski and Lavie, 2014) which combines unigram precision and recall with an internal alignment mechanism between the words in reference and hypothesis sentences. Unlike document-level BLEU which is unreliable when used to evaluate individual sentences (Song et al., 2013), METEOR is a sentence level metric by design that can also account for paraphrasing and synonyms for a set of languages including English, French and German. The highest achievable score for both metrics is 100.

Translation is a one-to-many problem in the sense that a single source sentence may have infinitely many acceptable human translations. The automatic metrics are by no means capable of fully handling such variabilites in the outputs but this can be achieved to some extent by using multiple reference sentences. Although BLEU and METEOR support multi-reference evaluation, very few datasets provide them for their test sets.

Manual Human Evaluation

Manual evaluation through human annotators is the primary evaluation method considered in the news translation shared task yearly held under the conference of machine translation (WMT). The type of manual evaluation preferred since 2017 (Bojar et al., 2018) is called “direct assessment” (DA) (Graham et al., 2017) where human annotators are presented with an MT output and its associated reference and asked to score the quality of the translation using a $[0, 100]$ scale. The collected annotations are then standardized within each annotator and then across all annotators to obtain an overall score for each system. A clustering based on significance test is finally performed to rank the systems.

3.6 Translation Decoding

Once an NMT model is trained, the translations for new sentences are generally decoded using the *greedy search* or the *beam search* (Graves, 2013; Boulanger-Lewandowski et al., 2013; Sutskever et al., 2014) algorithms. These search algorithms iteratively explore the search space to find the most likely translation for a given input, based on the log-likelihood estimate of the model. In this thesis, I always use the beam search which, given an input, proceeds as follows: the decoding first starts with an empty hypothesis. At timestep $t = 1$, we expand the empty hypothesis with every possible word in the target vocabulary $\mathbb{T}$ resulting in a list of $|\mathbb{T}|$ partial hypotheses. This list of partial hypotheses is called the *beam*. Before reiterating the same procedure for $t > 1$, beam search computes the log-likelihood of each hypothesis in the beam and prunes the beam to top k most likely hypotheses. The search stops when the <eos> token is generated for all k hypotheses in the beam. The size of the beam k is a predetermined hyperparameter usually ranging between 2 and 20. The greedy search is a special case of the beam search where only the most likely hypothesis is kept ($k = 1$) at each iteration.

Ensembling

Ensembling is a technique that allows averaging the predictions of an arbitrary number of models during the inference step. In the context of DNNs, training the same model multiple times with different random initializations and averaging their decisions often leads to substantial performance improvements. A common way of ensembling in NMT is to run the beam search algorithm on a set of trained models in a synchronized way and sum their log-likelihoods at each decoding step t. Sutskever et al. (2014) demonstrated that this improves over their single best NMT by 2.7 and 4.2 BLEU scores for a two-model and a five-model ensemble, respectively.

3.7 Summary

In this chapter, I introduced the task of machine translation along with the prominent approaches currently used in the field such as PBMTs and NMTs. I specifically focused on the latter as it is the fundamental framework that our multimodal translation approaches will be based on. After explaining in detail each component of NMTs such as the encoder, the decoder and the attention mechanism, I briefly described the beam-search algorithm and the commonly used translation evaluation methods. We now have the necessary background to start discussing multimodal machine translation.

CHAPTER 4

Multimodal Machine Translation

Human beings interact with their surrounding world mostly through visual, auditory and tactile sensory modalities. Language is often communicated over these sensory channels and perceived as a visual, auditory and tactile stimuli when looking at a word depicted in a traffic sign, listening to a conversation or reading a book written using the Braille system, respectively. Besides being able to handle each sensory modality in an isolated way, humans also develop a complex ability of integrating multiple modalities for efficient perception and decision making (Stein et al., 2009), including uncertainty reduction (Ernst and Banks, 2002). Computational language understanding also benefits from multimodality in ways similar to human perception. Silberer and Lapata (2012) showed that for semantic tasks such as word association and similarity, the joint modeling of linguistic and perceptual information correlates with human judgments better than late fusion of independent representations. Recent attempts at audio-visual speech recognition are forms of uncertainty reduction where noisy speech utterances are successfully transcribed by lip-reading from the video stream (Chung et al., 2017).

It is not a surprise that language understanding is at the heart of MT which requires inferring the meaning of a sentence in one language and transferring that meaning to another language. State-of-the-art approaches in NMT successfully leverage the *distributional hypothesis* (Firth, 1957) through the use of word embeddings and achieve meaning induction abilities solely by being exposed to large amounts of parallel sentences. Rios Gonzales et al. (2017) show that without any kind of explicit supervision, an out-of-the-box NMT is able to reach an accuracy of 70% for a word sense disambiguation (WSD) task in two different languages. However, there are many situations where purely distributional evidence is not sufficient to correctly translate a sentence. Consider the case where the translation of a sentence depends on the resolution of an anaphora with the antecedent being in the previous sentence or translating from a gender-neutral language to another one that has grammatical gender. Contextual (or large-context) MT is

specifically interested in solving the former problem by integrating cross-sentence information from neighboring sentences, paragraphs or even external linguistic resources (Tiedemann and Scherrer, 2017; Voita et al., 2018; Bawden, 2018). The grammatical gender problem however, can be solved by neither a human nor an MT system[1] without any additional context. What is worse for the MT system is how its word choices would be affected by the intrinsic gender bias of the training set (Prates et al., 2019), a major concern for language understanding methods based on word embeddings (Bolukbasi et al., 2016; Caliskan et al., 2017).

Multimodal machine translation (MMT) aims to provide a generic framework where the translation task is supported by auxiliary modalities such as vision and/or audio. Besides the aforementioned ambiguity issues, the successful integration of additional modalities can also be useful to improve the robustness of the system to noise, which can manifest itself as spelling mistakes or missing input words. The research efforts in MMT has so far been conducted on the Multi30K dataset (Elliott et al., 2016) which contains multilingual image descriptions and their translations. A yearly evaluation campaign has been held around the dataset to foster research on MMT (Specia et al., 2016; Elliott et al., 2017; Barrault et al., 2018). In this chapter, I first introduce the dataset and the evaluation campaign and then provide a review of the current state-of-the-art in MMT. I also briefly describe our contributions – which will be further detailed in next chapters – and conclude with a quantitative comparison of the described approaches in terms of MT evaluation metrics.

4.1 Multi30K Dataset

Multi30K (Elliott et al., 2016) is currently the prominent dataset used for MMT research. The dataset is derived from the Flickr30K dataset (Young et al., 2014) of image descriptions where five English descriptions were crowd-sourced for each of the 31014 images. In order to construct a parallel translation corpus with associated images, one of the five descriptions was professionally translated to German by human translators (Elliott et al., 2016). Although the translators were originally given the English sentence without the image, Frank et al. (2018) later collected "image-aware" post-edits from another human translator for the development and test set references. The dataset is later extended to include French (Elliott et al., 2017) and Czech (Barrault et al., 2018) translations, leading to 31014 English→German, English→French and English→Czech translation pairs with English sentences shared across all pairs. Unlike the original German translations, the French and Czech annotators were also given the described images as a visual cue.

[1]Google Translate palliated this problem by suggesting alternative translations to the user.

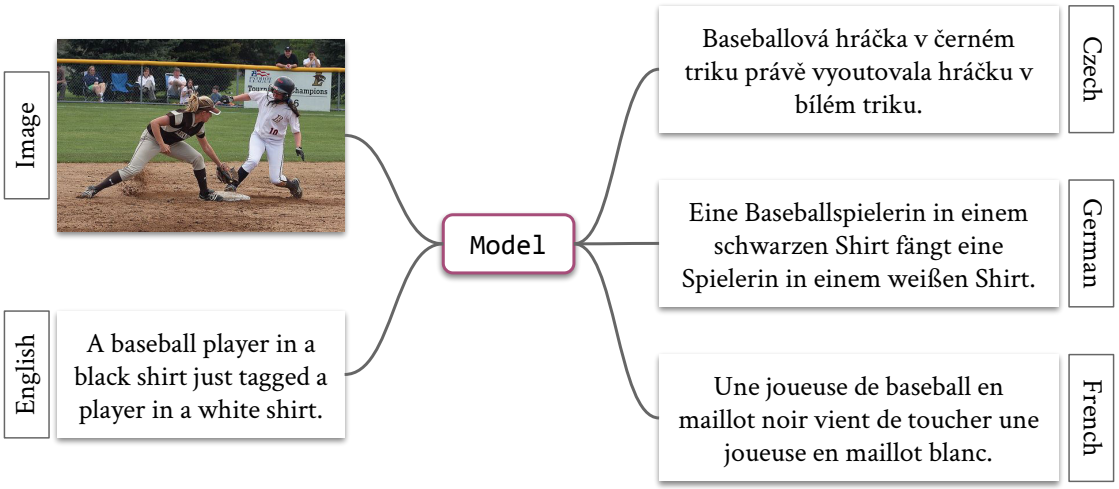


Figure 4.1: Bilingual subtasks of the shared task on MMT: An English→TRG model receives the image and the English sentence to be translated into TRG.

4.1.1 Shared Task on MMT

Multi30K is the primary training resource provided by the shared task on MMT, which is an evaluation campaign held under the Conference of Machine Translation (WMT) between 2016 and 2018 (Specia et al., 2016; Elliott et al., 2017; Barrault et al., 2018). Each year, a new language pair was added to the official evaluation campaign leading to three independent MMT subtasks in 2018, namely, English→German, English→French and English→Czech (Figure 4.1). A multimodal, multilingual subtask was also proposed in 2018 with the aim of designing a many-to-one MT system that considers the image and its English, French and German descriptions to perform translation into Czech. The use of additional resources such as MT and image captioning datasets is often encouraged and the submissions that use them are tagged as "unconstrained". At the end of the submission period, all participating systems are evaluated with METEOR and BLEU (section 3.5, p. 41) with METEOR being the primary metric. In 2017 and 2018, a human evaluation (section 3.5, p. 41) was also conducted using the direct assessment approach extended with the described images. In this thesis, we are solely interested in "constrained" English→German and English→French tasks.

Test Sets

Each year, a new test set is published to evaluate the performance of participating systems on unseen data. After the evaluation period, the references of the new test sets are disclosed so that researchers are able to evaluate their systems on them as well. One exception to that is the latest `test2018` set which is kept undisclosed for continuous

	English		German		French		
Split	Words	Avg. Len	Words	Avg. Len	Words	Avg. Len	Sents
train	380K	13.1	364K	12.6	416K	14.4	29000
val	13.4K	13.2	13.1K	12.9	14.6K	14.4	1014
test2016	13.0K	13.1	12.2K	12.2	14.3K	14.2	1000
test2017	11.4K	11.4	10.9K	10.9	12.8K	12.8	1000
testcoco	5.2K	11.4	5.2K	11.2	5.8K	12.5	461
Total	423K	13.0	405K	12.5	464K	14.3	32475

Table 4.1: Tokenized word and sentence statistics for Multi30K.

	English		German		French	
	Sents (%)	Words (%)	Sents (%)	Words (%)	Sents (%)	Words (%)
test2016	11.8	1.0	23.8	2.5	12.3	1.0
test2017	15.5	1.7	31.7	3.6	13.8	1.3
testcoco	11.1	1.1	34.5	3.6	16.1	1.5

Table 4.2: OOV statistics for Multi30K test sets. Sentence percentages reflect the percentage of sentences containing at least one OOV word.

MMT evaluation through an online competition server[2]. Apart from the yearly test sets, a test set called testcoco was published as a more challenging secondary test set in 2017 (Elliott et al., 2017). testcoco contains 461 carefully selected image-sentence pairs that potentially include ambiguous verbs having multiple senses. Specifically, it contains one to three samples per sense per verb for 56 verbs in total. For example, the following two senses of the English verb "to pass" require different verbs when translating to French: "a vehicle *passing (dépasser)* another vehicle" and "a vehicle *passing (traverser)* over a bridge". A – visually grounded – verb sense disambiguation can be helpful when translating this test set.

Dataset Statistics

I provide several sentence level and corpus level statistics for the English, German and French sentences of Multi30K in Table 4.1. These statistics are collected on tokenized and lowercased sentences, following the experimental framework of the thesis (section 5.2, p. 59). We notice that the sentences are quite short containing ∼14 words on average across all languages. The descriptive nature of the sentences turns out to be a limiting factor in terms of syntactic and semantic diversity: 16.7% and 7.2% of English training set sentences start with the bigram "a man" and "a woman", respectively. In overall, with only 29K sentences available for training, the dataset is smaller (and also simpler in

[2] https://competitions.codalab.org/competitions/19917

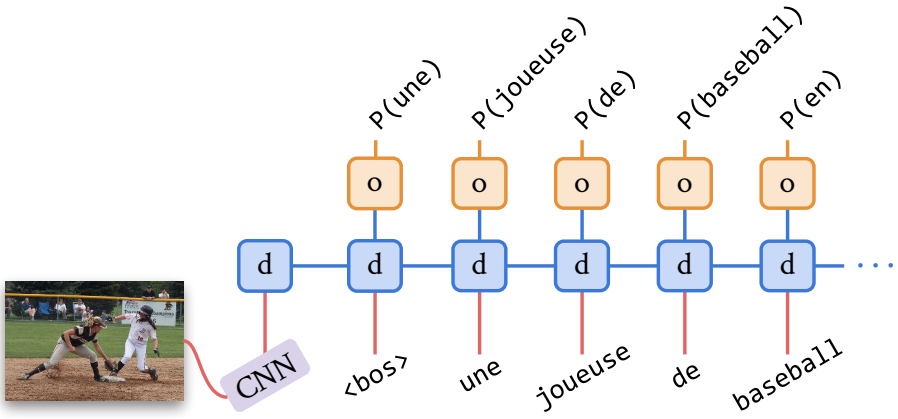


Figure 4.2: Show and tell captioning system (Vinyals et al., 2015): The decoder is an LSTM which receives a visual feature vector as its very first input.

terms of sentence structure) than commonly used MT datasets ranging from hundreds of thousands to hundreds of millions of sentences (Bojar et al., 2018). Table 4.2 provides OOV statistics for Multi30K test sets. With its compound words and rich morphology, it is not surprising that the German test sets are the most affected ones with more than $1/3$ of `test2017` and `testcoco` sentences containing at least one OOV word.

4.2 State-of-the-art in MMT

In this section, I review the current state-of-the-art in MMT by categorizing the approaches with respect to the type of visual features they integrate. First, I start with the models that make use of the global visual features and then move on to MMTs that incorporate convolutional (spatial) features. Although the main focus will be on neural approaches, prominent non-neural works are also described to some extent. Regardless of the type of feature involved, the majority of neural MMTs are inspired by previous works in neural image captioning (NIC) (Mao et al., 2015; Vinyals et al., 2015; Xu et al., 2015b). Unless otherwise stated, all models use features extracted from CNNs trained on ImageNet (Deng et al., 2009) image classification task (section 2.6.3, p. 27).

4.2.1 Global Visual Features

Although global features are spatially unaware and highly optimized for the initial task that they were trained for, notable works in NIC (Mao et al., 2015; Vinyals et al., 2015) successfully leveraged these features to generate natural language descriptions for images (Figure 4.2). Consequently, this type of feature turned out to be attractive in MMT research as well, where they have been shown to be beneficial to some extent.

Grounded Encoders & Decoders

The simplest way of leveraging the visual information in NMT consists of conditioning the encoders and/or the decoders with visual feature vectors: Calixto et al. (2016) and Libovický et al. (2016) use 4096-dimensional FC_7 features extracted from a VGG CNN (Simonyan and Zisserman, 2014) to **initialize** the hidden state of the recurrent **decoder**. Many extensions and refinements have been further proposed by concurrent works in 2017: Ma et al. (2017) initialize both the **encoder** and the **decoder** with 2048-dimensional "average pooled" feature vector of ResNet-50 (He et al., 2016) while Madhyastha et al. (2017) draw a comparison between the "average pooled" feature vector and the 1000-dimensional final probability vector of ResNet-152 (He et al., 2016) in similar encoder-decoder initialization scenarios. The authors also experiment with **additive** interaction between the feature vector and the **source embeddings** and find out that the probability vectors perform slightly better than the "average pooled" ones. Zheng et al. (2018) revisit **decoder initialization** and apply reinforcement learning techniques to fine-tune the model parameters with the objective of directly maximizing the BLEU score. They report that when combined with scheduled sampling (section 3.4.2, p.35), the fine-tuning yields BLEU improvements for NMT but the gains do not apply to MMT.

A slightly different grounding method is proposed by Huang et al. (2016) which consider the global feature vector as a "visual token" that can be **prepended** (or **appended**) to the sequence of source word embeddings. This implicitly allows the language attention mechanism to attend to visual information as they are made part of the source sequence. In essence, the proposed method is nothing more than a reiteration of Vinyals et al. (2015) (Figure 4.2) at encoder side. They further extend their approach by feeding the full image to a pre-trained object detection CNN (Girshick et al., 2014) to get four region proposals (*i.e.* bounding boxes) that contain salient objects. In addition to the global feature vector extracted from the full image (using a VGG CNN), they extract four more feature vectors for the proposed regions. A total of five visual vectors are then prepended to the source embedding sequence. In a similar vein, Calixto and Liu (2017) and Calixto et al. (2017a) simultaneously **prepend** and **append** the visual feature vector to the source sequence to ensure that the bidirectional encoder always processes the image as the first element. They also combine this with encoder and/or decoder initialization. Finally, Grönroos et al. (2018) experiment with RNN and Transformer (Vaswani et al., 2017) based NMTs by incorporating the visual feature in many ways such as **prepending** it, **multiplying** the embeddings with it (Caglayan et al., 2017a) or using it as a **gate** before the output layer to visually modulate the probability distribution over target words. More interestingly, they explore global visual features extracted from many different CNNs trained for scene recognition, action recognition and object detection. However, they obtain little

to none improvement from the visual modality and discover that when the models are given a mean feature vector for every sample, the translations do not deteriorate.

Multi-task Learning (MTL). A radically different encoder grounding technique is the *Imagination* (Elliott and Kádár, 2017) which is a one-to-many MTL architecture that shares the sentence encoder across a translation task T and a visual prediction task V. The latter aims to reconstruct the global visual feature vector from the "average pooled" source sentence encoding $\mathrm{H}_{\mathrm{AVG}}$ using a non-linear FC layer (Chrupała et al., 2015). A margin-based loss is used for the visual task to minimize the cosine distance between the true feature vector $\boldsymbol{f}$ and its reconstruction $\widehat{\boldsymbol{f}}$ while pushing away the latter from the "contrastive" global features sampled from the rest of the minibatch. The MTL loss is defined as the convex combination of the NMT loss and the visual loss:

$$
\begin{aligned}
\mathcal{J} &= \lambda\,\mathcal{L}_T + (1-\lambda)\mathcal{L}_V \\
\mathcal{L}_V &= \textstyle\sum_{\boldsymbol{f}'\neq\boldsymbol{f}} \max\Big(0, \alpha - \mathrm{distance}(\widehat{\boldsymbol{f}}, \boldsymbol{f}) + \mathrm{distance}(\widehat{\boldsymbol{f}}, \boldsymbol{f}')\Big) \\
\widehat{\boldsymbol{f}} &= \tanh(\boldsymbol{W}_v\,\mathrm{H}_{\mathrm{AVG}} + \boldsymbol{b}_v)
\end{aligned}
$$

The model is flexible in the sense that each task can be independently pre-trained on external resources and plugged into the model afterwards. Moreover, the visual features are not needed at test time as they are only used during training for grounding the shared encoder. The authors report improvements over their baseline especially in the constrained setup but when the NMT is pre-trained with additional data, the improvements do not seem to hold. Later on, Helcl et al. (2018) apply the same idea to a Transformer based NMT and show slight improvements over their baseline. Finally, Zhou et al. (2018) incorporate an auxiliary attention mechanism over the source sentence where the visual feature vector is used as the query to the attention. The margin-based loss now minimizes the distance between the true feature vector $\boldsymbol{f}$ and the output of the newly added attention layer instead of the reconstructed feature vector as in the original formulation.

Other Approaches

In this section, I briefly describe hybrid approaches based on reranking, retrieval and system combination. These approaches are often multi-stage in the sense that they consist of multiple submodels attached together in different ways. One of the earliest reranking based approaches is Shah et al. (2016) where the authors train a PBMT system and integrate the 1000-dimensional probability vector extracted from a CNN as additional scores for reranking 100-best list of translation hypotheses. Specifically, they consider each probability in the vector as a feature function for which a coefficient is estimated

during the tuning step. Their choice of probability features is motivated by the hypothesis that using likelihood of ImageNet objects appearing in the image may be more helpful than the penultimate layer features for MMT. This visual reranking yields very slight improvements over their PBMT baseline. In a more recent work, Lala et al. (2018) show that the 20-best translation candidates obtained from an NMT system actually contain high quality translations that potentially allow 10% absolute METEOR improvement. In order to select these candidates, they design a novel multimodal WSD system based on ResNet-50 global visual features and rerank their n-best list of translation candidates with scores assigned by the WSD system. However, they conclude that the Multi30K dataset do not significantly benefit from the proposed approach.

As for the retrieval based approaches, Duselis et al. (2017) and Gwinnup et al. (2018) consider the image as the driving modality for MMT instead of the language input. To this end, they train an image captioning system to generate candidate captions in the target language for each image. They utilize two encoders based on pre-trained FastText word embeddings (Bojanowski et al., 2017) to encode a source sentence and the candidate target captions obtained from the captioning system. After learning a mapping function between the source sentence space and the target caption space, they retrieve the target caption closest to the source caption in the learned mapping space. Finally, Zhang et al. (2017) propose a combined way of using retrieval and reranking. For a given sentence-image pair, they first retrieve a set of similar images from the training set based on the euclidean distance between the global visual features. The target sentences associated with the retrieved images are considered as candidate translations. They learn a visually guided word-to-word alignment function between source words and the candidate target words and use this function to select the most probable target word for each source word in the sentence. The 10K-best list of their PBMT is reranked with scores provided by a bidirectional NMT which receives the concatenation of the source words and the aligned target words. The authors report that pure reranking substantially improves the translation scores but the multimodal candidate word selection method shows no benefit.

4.2.2 Spatial Features

We now turn our attention to the second line of work in MMT that aims to integrate convolutional features into NMT. Unlike global features which provide a single vectorial representation, the spatial axis of convolutional features has the potential to allow an evolving integration scheme that fits within the iterative nature of encoders and/or decoders. However, these features are relatively less explored than global ones for MMT probably because of the challenges behind the design of multimodal fusion strategies that can take into account their representational complexity in an efficient way.

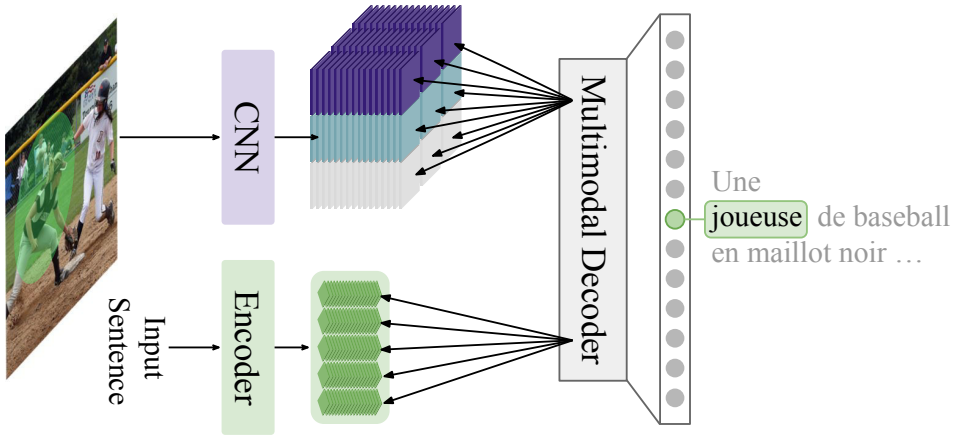


Figure 4.3: Multimodal decoder with visual attention: At decoding timestep $t = 2$, the hypothetical decoder correctly generates "joueuse (female player)" instead of "joueur" by integrating the image information.

Decoder-based Multimodality

Inspired from the success of visual attention in image captioning (Xu et al., 2015b), the majority of the previous work in "spatial MMTs" consider extending the attentive NMT with an auxiliary visual attention mechanism. A "black box" depiction is given in Figure 4.3 where the top and bottom parts correspond to visual and language attention mechanisms, respectively. To this end, Calixto et al. (2016) propose to extend the CGRU decoder (section 3.4.4, p. 38) with a visual attention layer $\mathbf{a}^{\mathrm{V}}$ that receives the spatial features $\mathbf{V}$. The final context $\boldsymbol{c}_1$ – which becomes the input to the second GRU – is then defined as the concatenation of language and visual contexts. The following summarizes the multimodal CGRU for the first decoding timestep:

$$\boldsymbol{c}_1 = \begin{bmatrix} \boldsymbol{c}_1^{\mathrm{L}} \\ \boldsymbol{c}_1^{\mathrm{V}} \end{bmatrix} = \begin{bmatrix} \mathbf{a}^{\mathrm{L}}\left(\mathbf{H}, \boldsymbol{h}_1'\right) \\ \mathbf{a}^{\mathrm{V}}\left(\mathbf{V}, \boldsymbol{h}_1'\right) \end{bmatrix} \tag{4.1}$$

Later on, Calixto et al. (2017b) apply the "gating scalar" from Xu et al. (2015b) with the purpose of scaling the visual context vector $\boldsymbol{c}^{\mathrm{V}}$ based on the hidden state of the decoder:

$$\beta = \sigma\left(\boldsymbol{W}_g \boldsymbol{h}_1' + \boldsymbol{b}_g\right)$$
$$\boldsymbol{c}_1^{\mathrm{V}} = \beta\left(\mathbf{a}^{\mathrm{V}}\left(\mathbf{V}, \boldsymbol{h}_1'\right)\right)$$

Although both models perform equally well compared to their baseline, the authors report that the latter model learns to activate the gate for visually depicted concrete nouns. Libovický and Helcl (2017) propose two multimodal attention variants, namely, the "flat" and the "hierarchical" attention. The flat attention combines the textual and

the visual encodings along the time and spatial axes to form a flat multimodal sequence $\mathbf{M} = \left[\mathbf{H} \in \mathbb{R}^{S \times c}; \mathbf{V} \in \mathbb{R}^{K \times c}\right]$ where $\mathbf{M} \in \mathbb{R}^{(S+K) \times c}$. Here, S denotes the number of words in the source sentence while K would be 64 for convolutional features with 8x8 spatial resolution. This new sequence replaces the text-only input to the attention layer originally found in the CGRU decoder. The final context is then given by $\boldsymbol{c_1} = \mathbf{a}\left(\mathbf{M}, \boldsymbol{h'_1}\right)$. On the other hand, the hierarchical attention follows the dedicated attention formulation of Calixto et al. (2016) but instead of concatenating the individual contexts, it utilizes a new "hierarchical" attention layer $\mathbf{a}^{\mathrm{H}}$ on top:

$$\begin{aligned}
\boldsymbol{c}_1^{\mathrm{L}} &= \mathbf{a}^{\mathrm{L}}\left(\mathbf{H}, \boldsymbol{h'_1}\right) \\
\boldsymbol{c}_1^{\mathrm{V}} &= \mathbf{a}^{\mathrm{V}}\left(\mathbf{V}, \boldsymbol{h'_1}\right) \\
\boldsymbol{c_1} &= \mathbf{a}^{\mathrm{H}}\left(\left[\boldsymbol{c}_1^{\mathrm{L}}; \boldsymbol{c}_1^{\mathrm{V}}\right], \boldsymbol{h'_1}\right)
\end{aligned}$$

The authors show that the hierarchical attention performs better than the flat one although it can not surpass their baseline.

Finally, with Transformer NMTs (TFNMT) (Vaswani et al., 2017) becoming more and more popular, researchers started to explore the integration of spatial features into the TFNMT decoder as well. Arslan et al. (2018) extend the decoder with the separate attention mechanism of Calixto et al. (2016) and fuse the obtained modality contexts with addition instead of the concatenation (Equation 4.1) while Libovický et al. (2018) integrate their previous flat and hierarchical attention and propose two more variants, namely, the "parallel" and the "serial" attention. The parallel attention closely follows Arslan et al. (2018) while the serial one applies the language and the visual attention in a stacked way where the former produces the query vectors for the latter. Arslan et al. (2018) substantially improve over their baseline TFNMT in terms of BLEU but strangely report a very poor METEOR score. For Libovický et al. (2018), the parallel attention works best with moderate improvements over their baseline for all three translation pairs.

Encoder-based Multimodality

Besides the commonly explored decoder-based multimodal strategies, two encoder-based methods exist in the literature (Delbrouck and Dupont, 2017a). The first one modulates the batch normalization (BN) layer (Ioffe and Szegedy, 2015) of the ResNet CNN which is used to extract the visual features. The BN layers are often placed after the convolutional layers to standardize the previous activations to zero mean and unit variance. At the same time, the layer also learns to rescale and reshift the normalized activations. Delbrouck and Dupont (2017a) propose to intervene at this specific step by injecting tiny variations to the learned mean and variance of a BN layer where the variations

are driven by the mean source sentence encoding H_{AVG}. This has the impact of modulating the feature maps extracted from the CNN in a learnable way where each feature map can be attenuated or amplified based on the source sentence. The reported results suggest that the method performs slightly inferior to the winning system from MMT17 (Caglayan et al., 2017a). Unfortunately, they do not compare the results with respect to their underlying baseline.

The second method couples the visual attention with the sentence encoder where the visual context is computed using the bidirectional hidden states. The visual contexts are then fused with the bidirectional states to yield a set of multimodal encodings. The CGRU decoder then applies its original attention layer on top of the new multimodal encodings. The authors only provide results for the combination of the first method above and this method where the performance slightly surpasses the same MMT17 system.

Multimodal Fusion

The models presented so far employ addition, concatenation or a hierarchical attention in order to fuse the individual contexts into the final multimodal one. Delbrouck and Dupont (2017b) take a different approach and apply multimodal compact bilinear pooling (MCBP) (Fukui et al., 2016) which is an efficient realization of the computationally expensive outer product. Assuming that the individual context vectors have the same dimensionality c, the outer product of two vectors $\boldsymbol{c}_1^{\text{L}}$ and $\boldsymbol{c}_1^{\text{V}}$ is a $c \times c$ matrix composed of elementwise multiplication of every element of the first vector with every element of the second. If one would like to project this matrix back into a c-dimensional space in order to feed it into the second GRU for example, the number of parameters in that layer (c^3) quickly reaches hundreds of millions. MCBP approximates this operation efficiently, showing notable improvements for visual question answering (VQA). In the context of MMT however, Delbrouck and Dupont (2017b) show that although MCBP seems to improve over concatenation (Equation 4.1), it is inferior to a simple elementwise multiplication between the contexts *i.e.* $\boldsymbol{c_1} = \boldsymbol{c}_1^{\text{L}} \odot \boldsymbol{c}_1^{\text{V}}$.

4.2.3 Our Contributions

I now briefly describe the contributions of this thesis by drawing parallels to the state-of-the-art. In Caglayan et al. (2016a), we simultaneously explore a reranking method and an end-to-end MMT approach. For reranking, we train a PBMT, a recurrent NMT and an NLM conditioned on global visual features (Aransa et al., 2015). The scores provided by the NMT and the visual NLM are used to rerank the 1000-best list of translation candidates obtained by the PBMT. The visual NLM produces a single LM score per candidate unlike the concurrent work of Shah et al. (2016) where each element of the visual feature vector is considered as an independent feature function. With slight gains over the baseline PBMT, the proposed model ranked first in MMT16 campaign (Specia et al., 2016). We do not further detail this approach but present it as a baseline whenever we provide a quantitative comparison across the state-of-the-art models. For the end-to-end MMT approach, we experiment with spatial features and propose the "multimodal attention" for the first time, concurrently with Calixto et al. (2016). We specifically explore a shared multimodal attention in contrast to their dedicated version. Later on, we extend our multimodal attention approach with different levels of sharing along with two multimodal fusion techniques, namely, the addition and the concatenation (Caglayan et al., 2016b). Finally, we propose several other refinements in Caglayan et al. (2018) where we mainly show that feature normalization is crucial for the visual attention to work correctly. **Chapter 7** details the multimodal attention experiments and provides quantitative and qualitative analyses using up-to-date models.

As for the global visual feature based MMTs, in Caglayan et al. (2017a) we explore several interaction methods within the framework of recurrent NMTs. Specifically, we start by replicating the RNN initialization techniques (Calixto and Liu, 2017) and then propose novel interaction schemes primarily based on elementwise multiplication of the visual features with several intermediate language representations of the NMT system. Our English→German and English→French submissions to MMT17 evaluation campaign (Elliott et al., 2017) ranked first with respect to automatic metrics. Moreover, our German system ranked first in human evaluation by significantly surpassing other submissions. We extensively cover these methods in **Chapter 6** and provide quantitative and qualitative analyses again with up-to-date retrained models.

Finally, following the source degradation protocols that we introduce in Caglayan et al. (2019a), we conduct several probing experiments in **Chapter 8** to shed a light on the visual awareness of our MMTs, as well as on the need for visual grounding in the context of Multi30K.

	Type	Feat.	B	M	Description
Caglayan et al. (2016a) †	RNN	Spatial	29.3	48.5	Shared Attention
Helcl and Libovický (2017)	RNN	Spatial	31.9	49.4	Hierarchical Attention
Calixto et al. (2016) †	RNN	Spatial	28.8	49.6	Separate Attention
Arslan et al. (2018)	TF	Spatial	41.0	53.5	Parallel Attention
Calixto and Liu (2017)	RNN	Global	36.9	54.3	Encoder Prep. & App.
Huang et al. (2016) †	RNN	Global	36.8	54.4	+ Regional Features
Calixto et al. (2017b)	RNN	Spatial	36.5	55.0	β-gated Attention
Calixto and Liu (2017)	RNN	Global	37.3	55.1	Decoder Init.
Elliott and Kádár (2017)	RNN	Global	36.8	55.8	Imagination (MTL)
Helcl et al. (2018)	TF	Global	38.8	56.4	Imagination (MTL)
Shah et al. (2016) †	PBMT	–	34.6	56.6	MMT16 Baseline
Shah et al. (2016) †	PBMT	Global	34.8	56.7	+Visual reranking
Caglayan et al. (2017a)	RNN	Spatial	37.0	57.0	Separate Attention
Libovický et al. (2018)	TF	Spatial	38.6	57.4	Parallel Attention
Caglayan et al. (2016a) †	PBMT	Global	36.2	57.5	Reranking (Visual NLM)
Caglayan et al. (2017a)	RNN	Global	38.2	57.6	Encoder Decoder Init.
Delbrouck and Dupont (2017a)	RNN	Spatial	40.5	57.9	BN + Enc. Attention
Grönroos et al. (2018)	TF	Global	45.1	–	Encoder Prep.

Table 4.3: (B)LEU and (M)ETEOR scores of state-of-the-art MMTs on `test2016` English→German. The highlighted system is unconstrained. The systems marked with (†) are re-evaluated with tokenized sentences. The descriptions refer to the techniques previously mentioned in this section.

4.2.4 Quantitative Comparison

I finalize this section with a quantitative overview of the current state-of-the-art in MMT for English→German `test2016` set as this is by far the most commonly used setup to report automatic metrics in literature. There exists an unfortunate discrepancy between the scores reported in MMT16 papers and the findings report (Specia et al., 2016) as the official evaluation for was performed using detokenized sentences. To synchronize the results across systems, I downloaded the submissions for MMT16 systems, tokenized and re-evaluated them accordingly. This results in an increase of around 2.5 and 4.5 points in BLEU and METEOR, respectively. Table 4.3 reports the final BLEU and METEOR scores for constrained systems in the literature along with the best unconstrained MMT18 submission (Grönroos et al., 2018) that may be considered as an upper bound. Although we leave the detailed analyses to the upcoming chapters, we can say that the results do not seem to suggest a distinctive boundary between the performance of global and spatial features.

4.3 Summary

In this chapter, I introduced the motivations behind MMT, described the closely associated Multi30K dataset, and provided an overview of the state-of-the-art. I broadly categorized the approaches into two groups based on the type of visual features they incorporate *i.e.* global visual features and spatial features. After briefly describing our contributions to MMT – that will be detailed in chapters 6, 7 and 8 – I summarized the current state-of-the-art in terms of automatic metrics. The next chapter details the common hyperparameters, the pre-processing workflow and the baseline NMT that will be extensively used throughout the remaining chapters.

CHAPTER 5

Experimental Framework

Throughout the course of MMT evaluation campaigns, we have progressively tuned our models each year to start with competitive baselines in the first place. Besides that, there has also been many changes in the way we have pre-processed the textual data and the CNN that we have used to extract visual features. This evolution makes it quite impossible to fairly compare our models among themselves and also to the current state-of-the-art. For this reason, the following chapters will present both the results obtained from the yearly evaluation campaigns and up-to-date results from systems specifically trained for this thesis. The latter systems use the same visual features, hyperparameters and pre-processing pipeline in order to ensure better comparability. In this chapter, I describe the experimental framework in detail and introduce the baseline NMT model on top of which the next chapters will be based on.

5.1 Software

As part of this thesis, I developed a high-level DNN Toolkit in Python called `nmtpy` with a focus on training language and vision related modalities and multimodal tasks. The first version of the toolkit (Caglayan et al., 2017b) was derived from the popular `dl4mt`[1] codebase and relied upon Theano (Theano Development Team, 2016) as the backend framework. The current version[2] which I extensively use in this thesis, is based on PyTorch (Paszke et al., 2017) framework. Although the fundamental model in `nmtpy` is the attentive NMT with CGRU decoder (Sennrich et al., 2017) (section 3.4.4, p. 38), the model agnostic API of the toolkit allows implementing and training different types of end-to-end DNNs pretty easily.

[1] https://github.com/nyu-dl/dl4mt-tutorial
[2] https://github.com/lium-lst/nmtpytorch

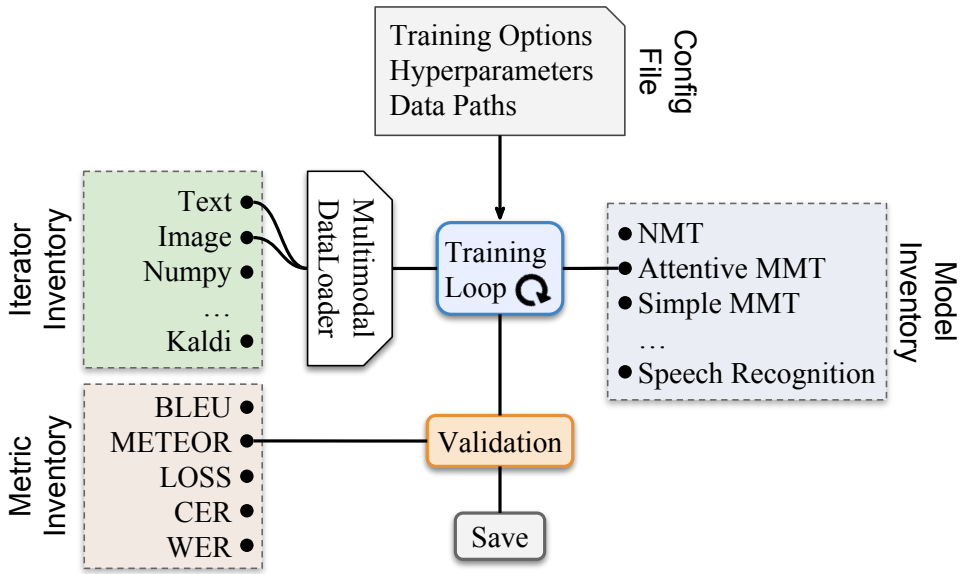


Figure 5.1: The training workflow of nmtpy.

Figure 5.1 summarizes the modular design and the training workflow of the toolkit: an experiment is fully defined by a configuration file which sets the training options, the paths to the relevant training and test set files, the specific model to be trained and its hyperparameters. Each input/output file is independently handled by the relevant iterator and a multimodal data loader coordinates these iterators to prepare minibatches of multimodal data. A model basically has to define a small set of methods to create the layers based on the received options and to realize the forward-pass. Finally, the training loop manages the whole training process where it also periodically evaluates the model using predefined metrics from the metric inventory.

Currently, nmtpy provides support for handling text files, arbitrary feature vectors, raw images and speech features in Kaldi format. As for the model inventory, it provides reference implementations for all the simple (chapter 6) and attentive MMT models (chapter 7) as well as a state-of-the-art speech recognition model and its multimodal extension (Caglayan et al., 2019b).

Besides my own works in MMT, nmtpy has also been successfully used by other researchers primarily for machine translation (Burlot et al., 2017; García-Martínez et al., 2017; Lala et al., 2018) and also for multimodal summarization (Libovický et al., 2018), phonemic transcriptions for text-to-speech (Vythelingum et al., 2018) and audio-visual dialog state tracking (Sanabria et al., 2019). The tool was also extensively used and developed by the "Grounded Sequence to Sequence Transduction" research group[3] within the Fifth Frederick Jelinek Memorial Summer Workshop in 2018.

[3] www.clsp.jhu.edu/workshops/18-workshop/grounded-sequence-sequence-transduction

5.2 Pre-processing

5.2.1 Image Features

We use a pre-trained ResNet-50 CNN (He et al., 2016) provided by `torchvision`[4] to extract visual features. For pre-processing the images, we resize the shortest edge to 256 pixels and then take a center crop of size 256x256. We extract spatial features of size 8x8x2048 from the final convolutional layer (`res5c_relu`) of the CNN. These spatial features are also the ones used to obtain the global 2048D `avgpool` features. In contrast, the shared task provides 14x14x1024 spatial features extracted from the second to last convolutional layer (`res4f_relu`) using images of size 224x224.

In chapter 6, we directly use normalized global features *i.e.* the L_2 norm of each feature vector is normalized to 1. In chapter 7, we provide an analysis of the impact of L_2 normalization and detail the experimental procedure there.

5.2.2 Text Processing

We use Moses (Koehn et al., 2007) scripts to lowercase, normalize and tokenize the sentences with aggressive hyphen splitting (`-a` parameter). For subword experiments, we use the BPE (Sennrich et al., 2016) algorithm (section 2.5.2, p.23) to create subword level vocabularies. For each language pair involved, we train a joint BPE model on the concatenation of the source and target training sentences. The number of merge operations is set to 10K for all language pairs.

5.3 Training & Evaluation

The set of common hyperparameters used throughout the thesis are given in Table 5.1. All models are trained for a maximum of 100 epochs. The model performance is evaluated at the end of each epoch based on METEOR score (Denkowski and Lavie, 2014) of the *val* set of Multi30K. If the METEOR score does not improve for ten epochs, the training is early-stopped (section 2.4.4, p. 14). In a similar way, the learning rate is halved if no improvement occurs for three consecutive epochs. We do not fix the seed of the random number generator and train all models *three* times with different random initializations. Once the training is over, we decode test set translations from each run separately using the beam search algorithm with a beam size of 12. Prior to evaluation, we recover all segmentation artifacts including the hyphen splitting and the BPE in order to ensure comparability across systems. We use the `multeval` tool (Clark et al., 2011) to compute

[4] https://pytorch.org/docs/stable/torchvision/models.html

Hyperparameter	Description & Value
Weight initialization	He et al. (2015)
Encoder type	Bi-directional GRU w/ 2 layers
	(initialized with **0** unless otherwise stated)
Decoder type	Conditional GRU w/ 2 layers
	(initialized with **0** unless otherwise stated)
Embedding Size	e = 200
RNNs hidden size	h = 320
Optimizer	Adam
	Batch size: 64, Learning rate: $4e-4$
Gradient clipping	if norm exceeds 1
L_2 regularization	$1e-5$
Dropout over	source embeddings with $p = 0.4$
	encodings $\mathbf{H}$ with $p = 0.5$
	the output logic with $p = 0.5$

Table 5.1: The common set of hyperparameters used in the thesis: the decoder embeddings are tied (Press and Wolf, 2016).

	English→German			English→French		
System	Vocab	BLEU	METEOR	Vocab	BLEU	METEOR
WRD→WRD	9800→18000	38.1	57.9	9800→11000	61.3	76.2
BPE→WRD	4800→18000	38.7	58.0	5300→11000	61.3	76.0
WRD→BPE	9800→6400	**38.9**	**58.4**	9800→5900	**61.4**	**76.4**
BPE→BPE	4800→6400	38.8	58.1	5300→5900	60.7	75.7

Table 5.2: NMT performance on `test2016` with different segmentation schemes.

tokenized BLEU and METEOR scores along with their means and standard deviations across three runs. We also rely on `multeval` to report statistical significance of the systems with respect to a designated baseline.

5.4 Baseline NMT

We conduct a preliminary experiment to select our baselines for English→German and English→French. Specifically, we test four systems that use word and subword vocabularies and report average BLEU and METEOR scores over three runs in Table 5.2. We observe that WRD→BPE systems with word-level source tokens and subword-level target tokens outperform other systems. Based on this, we select this system as the baseline architecture for the upcoming MMT experiments.

CHAPTER 6

Simple Multimodal Machine Translation

This chapter describes our simple MMT (SMMT) architectures that extend S2S NMTs by incorporating global visual features. These features are generally extracted from state-of-the-art CNNs (section 2.6.3, p. 27) primarily trained for large-scale vision tasks such as ImageNet image classification task (Deng et al., 2009). Global features can be thought as continuous bag of "latent concepts" where a linear layer applied on top, successfully classifies a given image into one of the thousand object categories. Although these vectors are highly tuned for the primary task that they were trained for, they were also showed to be effective for language related tasks such as bilingual lexicon induction (Kiela et al., 2015) and image captioning (Mao et al., 2015; Vinyals et al., 2015). Therefore, the majority of state-of-the-art MMTs (section 4.2.1, p. 47) rely on global features that are compact and thus easy to integrate into existing NMTs.

Our proposed SMMTs can be broadly divided into two categories: (i) initializing the sentence encoders and/or decoders similar to Calixto and Liu (2017), (ii) interacting the visual features and the intermediate language representations in the network in novel ways. We train the models on Multi30K dataset, following a fixed set of hyperparameters (Table 6.1) and the pre-processing pipeline previously described (section 5, p. 57). Finally, we conduct a quantitative analysis for English→German and English→French translation directions using corpus level and sentence level automatic evaluation and compare our systems to the current state-of-the-art in MMT. The chapter comprises the following work as well as unpublished extensions to it:

- Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 432–439.

Name	Symbol & Value
Source sentence length	S
Target sentence length	T
Embedding dim.	$e = 200$
RNNs hidden dim.	$h = 320$
Single textual encoding dim.	$c = 2h = 640$
All textual encodings	$\mathbf{H} \in \mathbb{R}^{S \times 640}$
Global visual feature	$\boldsymbol{f} \in \mathbb{R}^{2048}$
Transformed visual feature	$\boldsymbol{v} \in \mathbb{R}^{v}$

Table 6.1: Hyperparameters and intermediate dimensions for SMMTs: the dimension v of the transformed visual features depends on the type of interaction.

6.1 Methods

We first introduce SMMTs based on RNN initialization and then continue with elementwise interactions. A visual summary of all the models is sketched in Figure 6.1.

6.1.1 RNN Initialization

We define three models that aim to provide visual context to the encoders and/or decoders by initializing their hidden states with global visual features. The initialization based MMTs were first explored in Calixto et al. (2016) and later extended with other variants in Calixto and Liu (2017) (section 4.2.1, p. 48). Our models closely relate to these works with slight differences that will be detailed. Common to all three methods is the projection of the visual feature vector into the hidden space of the relevant RNN layer(s):

$$\boldsymbol{v} = \tanh\left(\boldsymbol{W}_f \, \boldsymbol{f} + \boldsymbol{b}_f\right) \qquad \boldsymbol{W}_f \in \mathbb{R}^{h \times 2048} \tag{6.1}$$

Encoder Initialization (`EINIT`)

Let us first remind the bidirectional sentence encoding step where the hidden states of both the forward and the backward GRUs are initialized with $\mathbf{0}$:

$$\mathbf{H} = [\boldsymbol{h_1}, \ldots, \boldsymbol{h_S}] = \mathsf{ENC}\left(\mathbf{X}, \boldsymbol{h_0} \leftarrow \mathbf{0}\right) = \begin{bmatrix} \overrightarrow{\mathsf{GRU}}\left(\overrightarrow{\mathbf{X}}, \overrightarrow{\boldsymbol{h_0}} \leftarrow \mathbf{0}\right) \\ \overleftarrow{\mathsf{GRU}}\left(\overleftarrow{\mathbf{X}}, \overleftarrow{\boldsymbol{h_0}} \leftarrow \mathbf{0}\right) \end{bmatrix}$$

We propose to initialize both forward and the backward GRUs with $\boldsymbol{v}$ (Figure 6.1, method 1) unlike Calixto and Liu (2017) where separate projections are preferred:

$$\overrightarrow{\boldsymbol{h_0}} = \overleftarrow{\boldsymbol{h_0}} = \boldsymbol{v}$$

Although not explicitly stated in the equation above, since our encoder is composed of two stacked GRU layers, the initialization is also applied to the forward and backward GRUs of the second encoder layer. We believe that providing the same projection to all RNN layers for all directions may be a more consistent signal as the visual context should be invariant to encoding direction. Sharing the projection is also parameter efficient.

Decoder Initialization (DINIT)

The decoder initialization (Figure 6.1, method 2) is the most commonly explored way of visual grounding in MMT probably inherited from early NMTs (Cho et al., 2014b; Sutskever et al., 2014) where the decoder is conditioned on a compressed source sentence representation through its initial state. Although this conditioning is no longer crucial with the introduction of the attention mechanism, the decoder layer(s) in NMTs are still initialized with some kind of information coming from the encoder (section 3.4.4, p. 38). The visual grounding method proposed here initializes the first GRU in the CGRU decoder with the projected visual features by setting $\boldsymbol{h}'_0 = \boldsymbol{v}$. To allow for a fair comparison between the NMTs and the MMTs explored in this thesis, we kept the decoder uninitialized in our baseline NMTs. This way, the proposed method does not have to override a textually initialized decoder. An alternative is to initialize the decoder in a multimodal fashion as in Calixto and Liu (2017) where $\boldsymbol{h}'_0$ is computed with an FF layer receiving $\boldsymbol{v}$ and H_{-1}.

Encoder & Decoder Initialization (EDINIT)

This method (Figure 6.1, method 1+2) constrains the network to learn a single representation that would satisfy all forward backward encoder layers as well as the first GRU in the decoder by using a single projection layer (equation 6.1). This is made possible since all five GRUs in our baseline have the same hidden state dimension h (Table 6.1):

$$\overrightarrow{\boldsymbol{h}_0} = \overleftarrow{\boldsymbol{h}_0} = \boldsymbol{h}'_0 = \boldsymbol{v}$$

Visual Beginning-of-Sentence (VBOS)

We previously saw that the target sequences in NMT training are prepended with a beginning-of-sentence token <bos> (section 3.4.2, p. 35). Once the model is trained and the parameters are fixed, this embedding – hence the initial input to the decoder – stays constant across different sentences that are translated. With this model, we propose to replace the static <bos> embedding with a dynamic one conditioned on the image information. The approach (Figure 6.1, method 5) is similar to Vinyals et al. (2015) in the sense that the decoder receives the feature vector as the first input but we further remove the

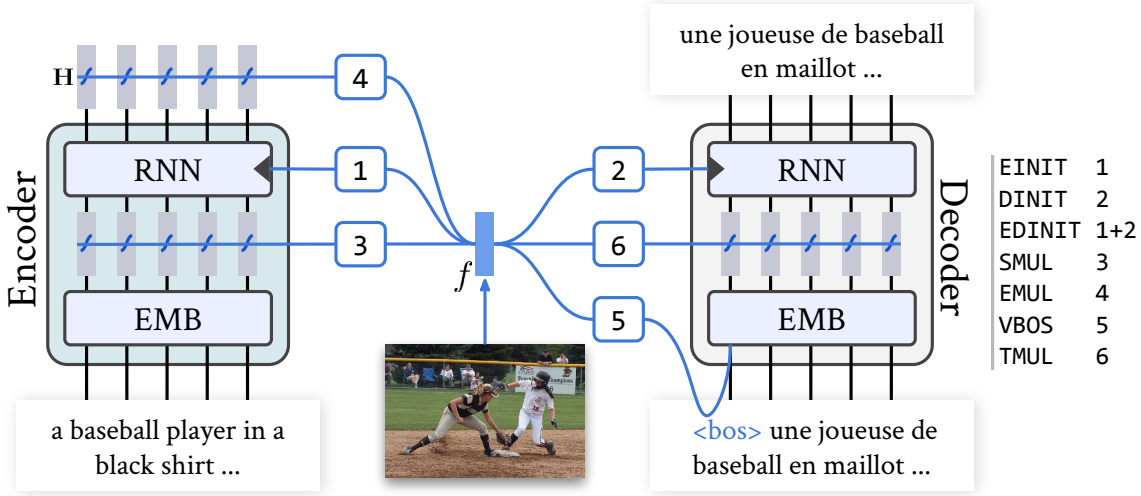


Figure 6.1: Visual summary of SMMT methods: $\boldsymbol{f} \in \mathbb{R}^{2048}$ is the feature vector extracted from ResNet-50 (He et al., 2016). Each model is characterized by one or more numbered paths as defined in the right side index.

explicit <bos> token so that the decoding is truly initiated with the visual context. The following shows the CGRU decoder logic for the first timestep and then modifies it accordingly to replace the <bos> with the linearly projected visual feature vector. Different from the RNN initialization methods (equation 6.1), we use a linear transformation here to comply with the dynamics of the word embeddings:

$$\boldsymbol{h}'_1 = \text{GRU}_1\left(\text{EMB}\left(\text{<bos>}\right), \boldsymbol{h}'_0 \leftarrow 0\right) \qquad \text{NMT}$$

$$\boldsymbol{h}'_1 = \text{GRU}_1\left(\left(\boldsymbol{W}_f\, \boldsymbol{f} + \boldsymbol{b}_f\right), \boldsymbol{h}'_0 \leftarrow 0\right) \qquad \boldsymbol{W}_f \in \mathbb{R}^{e \times 2048} \qquad \text{VBOS}$$

Although we have experimented with this model in the context of S2S multimodal speech recognition (Caglayan et al., 2019b), this is the first time that we explore it for MMT.

6.1.2 Elementwise Interaction

In Caglayan et al. (2017a), we propose three novel interaction types concerning source side and target side sentence representations. All variants employ multiplicative interaction between the language related representations and the transformed visual feature $\boldsymbol{v}$. The multiplicative interaction differs from additive interaction in terms of cross-modal nature of its backward dynamics: the gradient of the loss with respect to the language related vectorial representation is scaled by $\boldsymbol{v}$ and vice-versa ($\partial ab/a = b$) whereas the gradient with respect to the sum is directly passed along to the inputs of the sum. Fukui et al. (2016) show that the multiplicative interaction performs significantly better than the additive counterpart in VQA.

Let us denote the source and target sequences of embeddings by $\mathbf{X}=[\boldsymbol{x_1}, \ldots, \boldsymbol{x_S}]$ and $\mathbf{Y}=[\boldsymbol{y_1}, \ldots, \boldsymbol{y_T}]$, respectively. Below, we define the interactions in a single set of equations although they are never combined together during the experiments:

$$
\begin{aligned}
\boldsymbol{v} &= \tanh\left(\boldsymbol{W}_f\, \boldsymbol{f} + \boldsymbol{b}_f\right) \\
\mathbf{Y} &= [\boldsymbol{y_1} \odot \boldsymbol{v}, \ldots, \boldsymbol{y_T} \odot \boldsymbol{v}] && \text{TMUL} \\
\mathbf{X} &= [\boldsymbol{x_1} \odot \boldsymbol{v}, \ldots, \boldsymbol{x_S} \odot \boldsymbol{v}] && \text{SMUL} \\
\mathbf{H} &= \mathsf{ENC}\left(\mathbf{X}, \boldsymbol{h_0} \leftarrow \mathbf{0}, \boldsymbol{v}\right) = [\boldsymbol{h_1} \odot \boldsymbol{v}, \ldots, \boldsymbol{h_S} \odot \boldsymbol{v}] && \text{EMUL}
\end{aligned}
$$

TMUL multiplies the target embeddings with the visual vector (Figure 6.1, method 6) while the SMUL applies the same trick to source embeddings (Figure 6.1, method 3). EMUL integrates the multiplicative interaction into the output of the bi-directional encoder to modulate the source representations – on top of which attention will be applied in the decoder – with the visual vector $\boldsymbol{v}$ (Figure 6.1, method 4). The size of the projection matrix $\boldsymbol{W}_f$ is $\mathbb{R}^{e \times 2048}$ for embedding interactions TMUL and SMUL, and $\mathbb{R}^{c \times 2048}$ for the encoding interaction EMUL.

6.2 Results & Analysis

We report BLEU and METEOR scores for English→German and English→French translation directions on both test sets in Table 6.2. For **German**, we observe that the RNN initialization based variants EINIT, DINIT, EDINIT and the multiplicative interaction model TMUL obtain significantly different scores than the baseline on test2017 (p-value ≤ 0.05 according to multeval (Clark et al., 2011)). On average, TMUL seems the best performing model, reaching up to 0.7 point gains in both BLEU and METEOR. On test2016 however, the only significantly different systems are the EINIT and DINIT variants with up to 0.7 point gains in BLEU. The results are less promising for **French** where none of the systems achieve significantly different scores than the baseline. Still, we can say that the EDINIT, DINIT and TMUL systems – which are also ranked top three for German – are the ones that closely follow the baseline for test2017. The results suggest that the systems behave quite differently for German when compared to French. We now conduct a breakdown analysis based on sentence level METEOR scores to possibly gain some insights about this difference.

	test2016		test2017	
	BLEU	METEOR	BLEU	METEOR
	English→German			
NMT	38.9 ± 0.8	58.4 ± 0.3	32.1 ± 1.1	52.5 ± 0.7
EMUL	38.6 ± 0.4 (↓ 0.3)	58.1 ± 0.3 (↓ 0.3)	32.2 ± 0.4 (↑ 0.1)	52.3 ± 0.1 (↓ 0.2)
EINIT	39.6 ± 0.4 (↑ 0.7)	58.4 ± 0.2	32.9 ± 0.4 (↑ 0.8)	52.6 ± 0.2 (↑ 0.1)
VBOS	38.9 ± 0.1	58.3 ± 0.2 (↓ 0.1)	32.4 ± 0.9 (↑ 0.3)	52.8 ± 0.3 (↑ 0.3)
SMUL	39.0 ± 0.6 (↑ 0.1)	58.2 ± 0.4 (↓ 0.2)	32.4 ± 0.3 (↑ 0.3)	53.0 ± 0.1 (↑ 0.5)
EDINIT	39.0 ± 0.4 (↑ 0.1)	58.5 ± 0.3 (↑ 0.1)	32.9 ± 0.2 (↑ 0.8)	53.1 ± 0.3 (↑ 0.6)
DINIT	39.5 ± 0.1 (↑ 0.6)	58.6 ± 0.3 (↑ 0.2)	32.7 ± 0.5 (↑ 0.6)	53.1 ± 0.2 (↑ 0.6)
TMUL	38.8 ± 0.1 (↓ 0.1)	58.3 ± 0.2 (↓ 0.1)	32.7 ± 0.5 (↑ 0.6)	53.2 ± 0.1 (↑ 0.7)
	English→French			
NMT	61.4 ± 0.3	76.4 ± 0.2	54.4 ± 0.3	71.1 ± 0.2
EMUL	61.2 ± 0.2 (↓ 0.2)	76.2 ± 0.2 (↓ 0.2)	54.0 ± 0.7 (↓ 0.4)	70.6 ± 0.4 (↓ 0.5)
SMUL	60.9 ± 0.8 (↓ 0.5)	76.0 ± 0.4 (↓ 0.4)	53.9 ± 0.5 (↓ 0.5)	70.8 ± 0.3 (↓ 0.3)
EINIT	61.1 ± 0.2 (↓ 0.3)	76.0 ± 0.3 (↓ 0.4)	54.0 ± 0.1 (↓ 0.4)	70.8 ± 0.1 (↓ 0.3)
VBOS	61.4 ± 0.3	76.3 ± 0.2 (↓ 0.1)	54.1 ± 0.2 (↓ 0.3)	70.9 ± 0.3 (↓ 0.2)
EDINIT	60.8 ± 0.2 (↓ 0.6)	76.1 ± 0.1 (↓ 0.3)	54.1 ± 0.8 (↓ 0.3)	71.0 ± 0.4 (↓ 0.1)
DINIT	61.4 ± 0.3	76.4 ± 0.3	54.1 ± 0.2 (↓ 0.3)	71.0 ± 0.2 (↓ 0.1)
TMUL	61.1 ± 0.5 (↓ 0.3)	76.2 ± 0.2 (↓ 0.2)	54.2 ± 0.2 (↓ 0.2)	71.1 ± 0.3

Table 6.2: Combined SMMT results on test2016 and test2017: Highlighted scores are significantly different than the NMT (p-value ≤ 0.05). Ordered by test2017 METEOR.

6.2.1 Sentence Level Analysis

The protocol that we use for sentence level analysis is as follows: First, for each sentence of the test2017 set, we compute the METEOR scores obtained by the three independent runs of a given system. Second, we average these three scores to obtain a smoothed sentence level METEOR. Finally, for each MMT, we count the sentences which have a smoothed METEOR equal to, better than, or worse than the one obtained by the baseline NMT. In other words, for a given multimodal-monomodal translation pair, we completely disregard the absolute METEOR difference between them and discretize the evaluation into three bins of ties ($=$), wins ($>$) and losses ($<$).

Figure 6.2 shows the results for both language pairs. First, we notice that **French** systems behave very similarly to each other, with "losses %" higher than "wins %". Only the EDINIT system marginally differs in this aspect with 34.3% "losses" and 35.1% "wins". On the other hand, all **German** systems except the EMUL have more "wins" than "losses": The

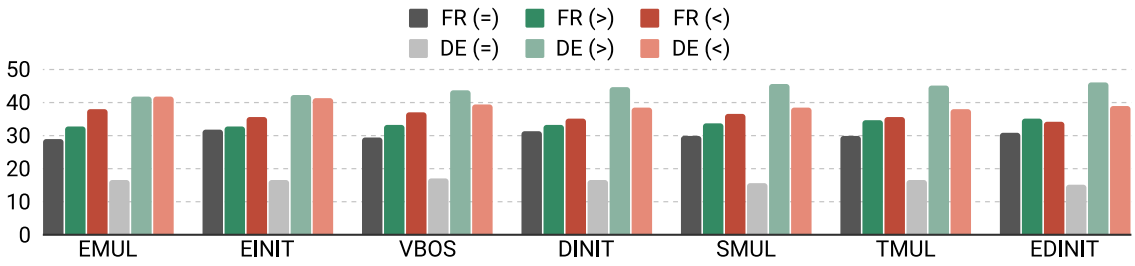


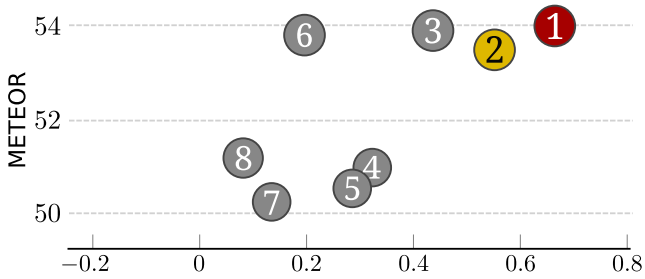
Figure 6.2: Sentence level METEOR breakdown for MMT systems: the results are ordered by German (DE) "wins - losses" gap.

EDINIT system improves %46.2 of the translations while deteriorating on 38.8%, exhibiting the largest "wins – losses" gap of 7.4%. Second, we observe a systematic difference between the languages in terms of the "ties": On average, 30.1% of multimodal **French** translations preserve their METEOR while for **German** this percentage drops to 16.1%. The ties are consistently stable across SMMT variants with a standard deviation of 1% and 0.6% for French and German, respectively. This shows that independent from the model type, there is always a nonnegligible portion of the test set for which any given model performs equivalent to the baseline NMT. The fact that the French portion is almost the double of its German counterpart raises another question: Is this difference related to the integration of visual modality or not? To understand this, we train a "control" NMT (still with three runs) and compute the same statistics for it by comparing its sentence level METEOR scores to the actual baseline. Surprisingly, we observe almost the same "ties" percentages for the "control" NMT: 29.9% for French and 16.8% for German. This strongly suggests that the French task is simpler than German since ∼1/3 of the test set consistently obtains equivalent sentence METEOR scores independent from the underlying conditions.

In overall, this fine-grained analysis corroborates the hypothesis that there is less room for improvement for the French task when compared to German. Although this is already obvious in terms of the corpus level results where French BLEU scores are ∼22 higher than German ones (Table 6.2), the breakdown analysis with discretized bins revealed interesting details about the characteristics of both NMTs and MMTs.

6.2.2 MMT17 Evaluation Campaign

In 2017, we participated to the shared task on MMT (Elliott et al., 2017) for both German and French translation directions. Back at that time, we mainly experimented with the DINIT, EDINIT, EMUL and TMUL variants presented in this chapter and submitted a 5-run ensemble of TMUL for German and a 6-run ensemble of mixed SMMTs for French, respectively. Our systems ranked first among 11 French and 16 German systems according to



(1) SMMT (Caglayan et al., 2017a)
(2) Imagination (Elliott and Kádár, 2017)
(3) Reranking (Zhang et al., 2017)
(4) NMT (Helcl and Libovický, 2017)
(5) SMMT (Calixto et al., 2017a)
(6) NMT (Caglayan et al., 2017a)
(7) Attention (Helcl and Libovický, 2017)
(8) Imagination (Elliott and Kádár, 2017)

Figure 6.3: Human judgment score vs METEOR for German MMT17 participants: Colors of the circles represent significantly different clusters based on Wilcoxon signed-rank test (p-value ≤ 0.05). Systems within a cluster are tied. The highlighted systems 2, 4 and 7 on the right are unconstrained. Figure adapted from Elliott et al. (2017).

`test2017` METEOR. Moreover, our multimodal German submission ranked first according to human evaluation (section 3.5, p. 41). Figure 6.3 plots the standardized human judgment score against METEOR for the top ranked constrained and unconstrained systems. The system (1) is our aforementioned `TMUL` ensemble while the system (6) is our 5-run NMT ensemble. Our winning ensemble also surpassed three unconstrained systems, namely, an *Imagination* MMT (2), a pure NMT (4) and an attentive MMT (7).

MMT17 vs. Retrained Systems

We now would like to compare the retrained systems for this thesis to the ones from MMT17 (Caglayan et al., 2017a). The new hyperparameters (section 5.4, p. 60) are substantially different than MMT17 as we now have a 2-layer encoder, 256D embeddings (instead of 128) and 320D recurrent layers (instead of 256). Also, we now use word→BPE vocabularies instead of the previous BPE→BPE ones. Table 6.3 provides average METEOR with standard deviation for the baseline NMT and `TMUL` systems. First of all, we can say that the new hyperparameters result in an average improvement of 0.9 points for **German** baseline. Second, the `TMUL` system brings up to 0.7 METEOR improvement both for MMT17 and the retrained systems. For **French** however, the 2 points gain (67.5→69.5) in MMT17 no longer holds for the retrained systems: The new baseline easily closes that gap and further reaches 71.1 with an overall improvement of ∼4 METEOR.

	EN→DE	EN→FR
NMT	$51.6 \pm 0.5 \Rightarrow 52.5 \pm 0.7$	$67.5 \pm 0.7 \Rightarrow 71.1 \pm 0.2$
TMUL	$52.2 \pm 0.4 \Rightarrow 53.2 \pm 0.1$	$69.5 \pm 0.7 \Rightarrow 71.1 \pm 0.3$

Table 6.3: `test2017` METEOR comparison of MMT17 systems to this thesis: the arrow ($\Rightarrow$) between the scores shows the transition from MMT17 to retrained systems. A vertical comparison reveals the multimodal improvements within each year.

System	German Rank (score)	French Rank (score)
SMMT (Caglayan et al., 2017a)	1 (0.67)	4 (0.22)
Reranking (Zhang et al., 2017)	3 (0.44)	1 (0.45)
SMMT (Calixto et al., 2017a)	5 (0.31)	3 (0.30)
NMT (Caglayan et al., 2017a)	6 (0.20)	8 (-0.08)

Table 6.4: Standardized human judgment scores for German and French: we compare the subset of the German systems (Figure 6.3) that also participated to French evaluation.

When we further compare the human evaluation rankings of German and French MMT17 submissions (Table 6.4), we notice how our French systems lag behind the other ones that were otherwise surpassed for English→German. The systems ranked 3rd and 5th for German move to the 1st and 3rd positions for French by obtaining almost the same standardized human judgment scores whereas our submissions obtain substantially lower scores compared to our German systems. The shift between same architectures trained for different languages is rather unexpected and is probably due to the differences between German and French hyperparameters back at that time, especially the ones related to dropout and L_2 regularization. Once again, This points out the importance of carefully selecting the underlying hyperparameters to avoid starting with a baseline that underfits or overfits to the training set.

6.3 Comparison to State-of-the-art

In this section, I compare our best SMMT systems to a selection of state-of-the art MMT systems including a competitive Transformer-based attentive MMT (Libovický et al., 2018). I evaluate the systems exactly the same way as the section 4.2.4 (p.55). According to the results in Table 6.5, our newly trained systems obtain the best BLEU and METEOR scores among the constrained systems, improving over our MMT17 systems as well. I report the relative gains (or drops) of each system with respect to the baseline MT reported in their works. For example, the difference between the baseline NMT and the `DINIT` model of Calixto and Liu (2017) is 2.8 points (52.3→55.1). These relative differences reveal a clear pattern among the current state-of-the-art in MMT: As researchers converge to better baselines, the apparent improvements due to multimodality tend to disappear. Our findings about the mismatch between our French MMT17 systems and the retrained ones also supports this view. We develop more insights about this aspect in chapter 8.

	System	BLEU	METEOR	Description
Calixto and Liu (2017)	RNN	36.9 (↑ 3.2)	54.3 (↑ 2.0)	Encoder Prep. & App.
Huang et al. (2016)	RNN	36.8 (↑ 2.0)	54.4 (↑ 2.3)	+ Regional Features
Calixto and Liu (2017)	RNN	37.3 (↑ 3.6)	55.1 (↑ 2.8)	DINIT
Elliott and Kádár (2017)	RNN	36.8 (↑ 1.3)	55.8 (↑ 1.8)	Imagination (MTL)
Helcl et al. (2018)	TF	38.8 (↑ 0.7)	56.4 (↑ 0.2)	Imagination (MTL)
Shah et al. (2016)	PBMT	34.8 (↑ 0.2)	56.7 (↑ 0.1)	Reranking (Visual NLM)
Libovický et al. (2018)	TF	38.6 (↑ 0.3)	57.4 (↑ 0.7)	Parallel Attention
Caglayan et al. (2016a)	PBMT	36.2 (0.0)	57.5 (↑ 0.1)	Multimodal NLM
Caglayan et al. (2017a)	RNN	38.2 (↑ 0.1)	57.6 (↑ 0.3)	EDINIT
This Chapter	RNN	38.8 (↓ 0.1)	58.3 (↓ 0.1)	TMUL
		39.0 (↑ 0.1)	58.5 (↑ 0.1)	EDINIT
		39.5 (↑ 0.6)	58.6 (↑ 0.2)	DINIT

Table 6.5: Comparison of state-of-the-art SMMTs on German test2016: TF stands for Transformer (Vaswani et al., 2017). We do not report ensemble results to ensure a fair comparison. The relative differences inside parentheses are with respect to the baseline MTs reported in those works. The results are sorted by METEOR.

6.4 Summary

In this chapter, I presented several SMMT systems which are MMTs that incorporate global visual features extracted from pre-trained CNNs. The chapter covers the systems proposed in Caglayan et al. (2017a) and adds two more SMMT systems to the inventory, namely, the SMUL and the VBOS variants. I provide quantitative results for the German and the French translation tasks of Multi30K dataset, using BLEU and METEOR scores on two different test sets. I further compare the systems retrained for this chapter to our winning submissions in MMT17. The main conclusions can be summarized as follows:

- We observe significant improvements in BLEU and METEOR for English→German – especially on test2017 – but the same does not hold for English→French.
- We conduct a sentence level analysis based on METEOR scores to break down the large baseline difference between German and French. The results corroborate the hypothesis that there is less room for improvement for French as the percentage of the test set consistently obtaining same METEOR across different NMT and MMT systems is 30% for French compared to 16% for German. In other words, French systems seem more stable and conservative in terms of the variability of the produced translations.

- Based upon a comparison between our MMT17 systems (Caglayan et al., 2017a) and the ones retrained in this chapter, we conclude that the RNN initialization based SMMTs along with the multiplicative `TMUL` variant exhibit moderate improvements for German regardless of the baseline performance. However, the significant multimodal improvements for French disappear with the retrained systems suggesting that the visual modality may be helpful only if the architecture has difficulty to fully exploit the textual information. We leave a quantitative and qualitative exploration of this aspect to chapter 8.

The next chapter explores a substantially different MMT paradigm equipped with a multimodal attention mechanism which exploits spatially aware convolutional features instead of the global visual features.

CHAPTER 7

Attentive Multimodal Machine Translation

The previous success of the attention mechanism led to the further exploration of the idea for multi-input and/or multi-output networks mostly in the context of multilingual NMT. Dong et al. (2015) and Zoph and Knight (2016) experimented with dedicated attention layers in one-to-many and many-to-one NMT systems respectively, whereas Firat et al. (2017) proposed a shared attention across multiple language pairs in a many-to-many framework. In overall, all these approaches seemed beneficial to translation performance according to the experimental results provided by the authors. However, the curious case of shared vs dedicated attention layers were not further explored in a comparative manner. Being a many-to-one framework with multiple input modalities involved to perform translation, MMT with visual attention lies at the intersection of the above approaches as well. Moreover, the aforementioned case of sharing the attention becomes much more interesting for MMT where the nature of the modalities are radically different: word representations and the corresponding encoder are jointly learned during the training while the visual representations are – generally – frozen and pre-trained for an external visual recognition task.

This chapter describes our efforts towards the design of attentive MMT (AMMT) architectures capable of integrating the visual modality through an additional visual attention module (Xu et al., 2015b). We begin by exploring a shared multimodal attention (Caglayan et al., 2016a) similar to Firat et al. (2017) and then manipulate it progressively to reach a completely dedicated variant along with different multimodal fusion strategies (Caglayan et al., 2016b). We compare all methods using a fixed set of hyper-parameters (Table 7.1) and the previously described pre-processing pipeline (section 5, p. 57). We finalize the chapter with a quantitative analysis on English→German and English→French translation tasks of Multi30K and also provide some qualitative insights about the characteristics of visual attention.

Name	Symbol & Value
Source sentence length	S
Target sentence length	T
Embedding dim.	$e = 200$
Encoder hidden dim.	$r = 320$
Single textual encoding	$c = 2r = 640$
All textual encodings	$\mathbf{L} \in \mathbb{R}^{S \times c}$
Spatial resolution	$K = 8 \times 8 = 64$
Raw visual features	$\mathbf{F} \in \mathbb{R}^{8 \times 8 \times 2048}$
Transformed visual encodings	$\mathbf{V} \in \mathbb{R}^{K \times c}$
Decoders hidden dim.	$d = 320$
Internal attention dim.	$a = d = 320$

Table 7.1: Hyperparameters and intermediate dimensions for attentive MMTs.

This chapter comprises the following published works:

- Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016a. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 627–633.
- Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016b. Multimodal attention for neural machine translation. *Computing Research Repository* arXiv:1609.03976.
- Ozan Caglayan, Adrien Bardet, Fethi Bougares, Loïc Barrault, Kai Wang, Marc Masana, Luis Herranz, and Joost van de Weijer. 2018. LIUM-CVC submissions for WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 603–608.

7.1 Revisiting the CGRU Decoder

I will first start by describing the decoder logic in the CGRU architecture in detail before extending it with the proposed multimodal attention mechanism. In the following, $\mathbf{L}$ and $\mathbf{V}$ denote the textual and the visual set of encodings, respectively. $\mathbf{L}$ is an alias for the usual set of bidirectional encodings $\mathbf{H}$ (section 3.4.4, p.38) while $\mathbf{V}$ represents the spatial features extracted from a pre-trained ResNet-50 CNN (He et al., 2016). We superscript all layers and transformations with V(isual) or L(anguage) to distinguish modality-specific constructs.

We start by naming the existing attention mechanism in the CGRU decoder the "language attention layer" and assign the symbol ATT^{L} to it. This layer computes the attention distribution over the language encodings $\mathbf{L}$ by using the hidden state $\boldsymbol{h}'_{\mathbf{t}}$ of the first decoder GRU as the query vector. Note that unlike the previous formulation (section 3.4.4, p. 38), here we separate out the context computation for reasons that will be clear once we introduce the multimodal fusion. The following equations summarize how to obtain the attention distribution $\boldsymbol{\alpha}^{\text{L}}_{\mathbf{t}}$ at decoding timestep t. We use a compact notation here to not explicitly define the attention scores for each source word position. This avoids cluttering the symbols with source position indices:

$$(\mathbf{L} \in \mathbb{R}^{S \times c}\,,\ \boldsymbol{W}^{\text{L}}_{e} \in \mathbb{R}^{c \times a}\,,\ \boldsymbol{h}'_{\mathbf{t}} \in \mathbb{R}^{1 \times d}\,,\ \boldsymbol{W}^{\text{L}}_{q} \in \mathbb{R}^{d \times a}\,,\ \boldsymbol{w}^{\text{L}}_{a} \in \mathbb{R}^{a \times 1})$$

$$\boldsymbol{\alpha}^{\text{L}}_{\mathbf{t}} = \text{ATT}^{\text{L}}\left(\mathbf{L}, \boldsymbol{h}'_{\mathbf{t}}\right) = \text{SOFTMAX}\left(\tanh\left(\mathbf{L}\,\boldsymbol{W}^{\text{L}}_{e} + \boldsymbol{h}'_{\mathbf{t}}\,\boldsymbol{W}^{\text{L}}_{q}\right)\boldsymbol{w}^{\text{L}}_{a}\right) \qquad \boldsymbol{\alpha}^{\text{L}}_{\mathbf{t}} \in \mathbb{R}^{S \times 1}$$

Once the attention distribution is computed, the language context $\boldsymbol{c}^{\text{L}}_{\mathbf{t}}$ is easily obtained with a matrix-vector product (equation 7.1). Finally, we linearly transform $\boldsymbol{c}^{\text{L}}_{\mathbf{t}}$ to make its size compatible with the input size d of the second GRU in the decoder (equation 7.2). This transformed context $\boldsymbol{i}_{\mathbf{t}}$ becomes the input to the second GRU:

$$\boldsymbol{c}^{\text{L}}_{\mathbf{t}} = \mathbf{L}^{\top}\,\boldsymbol{\alpha}^{\text{L}}_{\mathbf{t}} \qquad \boldsymbol{c}^{\text{L}}_{\mathbf{t}} \in \mathbb{R}^{c} \tag{7.1}$$

$$\boldsymbol{i}_{\mathbf{t}} = \boldsymbol{W}^{\text{L}}_{d}\,\boldsymbol{c}^{\text{L}}_{\mathbf{t}} \qquad \boldsymbol{W}^{\text{L}}_{d} \in \mathbb{R}^{d \times c} \tag{7.2}$$

$$\boldsymbol{h}''_{\mathbf{t}} = \text{GRU}_2\left(\boldsymbol{i}_{\mathbf{t}}, \boldsymbol{h}'_{\mathbf{t}}\right) \tag{7.3}$$

The rest of the computations follow the original CGRU formulations *i.e.* the probability of the next target token is computed with a deep output logic (section 3.4.4, p. 38). The language attention layer ATT^{L} is parameterized by the following transformations: $\{\boldsymbol{w}^{\text{L}}_{a}\,, \boldsymbol{W}^{\text{L}}_{e}\,, \boldsymbol{W}^{\text{L}}_{q}\}$. We further separate these three transformations into two groups where $\boldsymbol{W}^{\text{L}}_{q}$ is referred to as the "decoder-state" projection and $\{\boldsymbol{w}^{\text{L}}_{a}\,, \boldsymbol{W}^{\text{L}}_{e}\}$ are considered to be "modality-relevant" projections (Caglayan et al., 2016b).

7.2 Visual Attention

We denote the spatial features extracted from the pre-trained ImageNet CNN with the 3D tensor $\mathbf{F}$. Since we do not experiment with fine-tuning the CNN during MMT training, we extract the spatial features once for all the images in the dataset and plug these into our architecture afterwards as standalone features:

$$\mathbf{F} = \text{RESNET50}\left(\text{IMG}_{256 \times 256}\right) \qquad \mathbf{F} \in \mathbb{R}^{8 \times 8 \times 2048} \tag{7.4}$$

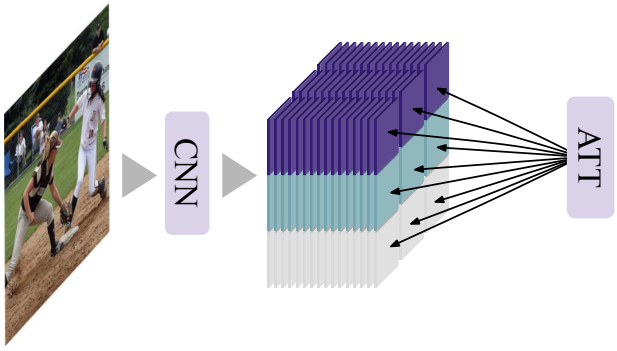


Figure 7.1: Spatial attention mechanism attends on the convolutional feature maps extracted from a raw image (Xu et al., 2015b).

Inside the network, a convolutional layer with c 1x1 filters is applied to the spatial features to make the feature dimension compatible with the language encodings $\mathbf{L}$. A FLATTEN operation is performed to flatten the spatial dimensions (8x8) of the feature tensor into 64 (feature vectors) on top of which a secondary attention mechanism can be applied without any change:

$$\begin{aligned} \mathbf{F}' &= \text{CONV}_{1\times1\times c}\left(\mathbf{F}\right) & \mathbf{F}' &\in \mathbb{R}^{8\times8\times c} \\ \mathbf{V} &= \text{FLATTEN}\left(\mathbf{F}'\right) & \mathbf{V} &\in \mathbb{R}^{64\times c} \end{aligned} \tag{7.5}$$

The type of visual attention explored so far in MMT is "spatial" (Xu et al., 2015b) in the sense that a probability mass is assigned to each position in the 8x8 grid of convolutional features (Figure 7.1). This way, the model is able to select "where" to attend in the image at each decoding timestep t. This formulation is quite similar to the language attention where a probability mass is assigned to each of the S hidden states produced by the encoder.

Let us now create a second attention layer ATT^{V} with another set of parameters $\{\boldsymbol{w}_a^{\text{V}}, \boldsymbol{W}_e^{\text{V}}, \boldsymbol{W}_q^{\text{V}}\}$ in order to implement the visual attention. Note how the two attention formulations are exactly the same except the number of feature vectors which is the number of words S and the spatial resolution $K = 64$ for the language and the visual attention, respectively:

$$\begin{aligned} &(\mathbf{V} \in \mathbb{R}^{64\times c}, \boldsymbol{W}_e^{\text{V}} \in \mathbb{R}^{c\times a}, \boldsymbol{h}'_{\mathbf{t}} \in \mathbb{R}^{1\times d}, \boldsymbol{W}_q^{\text{V}} \in \mathbb{R}^{d\times a}, \boldsymbol{w}_a^{\text{V}} \in \mathbb{R}^{a\times 1}) \\ &\boldsymbol{\alpha}_{\mathbf{t}}^{\text{V}} = \text{ATT}^{\text{V}}\left(\mathbf{V}, \boldsymbol{h}'_{\mathbf{t}}\right) \\ &\quad = \text{SOFTMAX}\left(\tanh\left(\mathbf{V}\,\boldsymbol{W}_e^{\text{V}} + \boldsymbol{h}'_{\mathbf{t}}\,\boldsymbol{W}_q^{\text{V}}\right)\boldsymbol{w}_a^{\text{V}}\right) \qquad \boldsymbol{\alpha}_{\mathbf{t}}^{\text{V}} \in \mathbb{R}^{64\times1} \\ &\boldsymbol{c}_{\mathbf{t}}^{\text{V}} = \mathbf{V}^{\top}\,\boldsymbol{\alpha}_{\mathbf{t}}^{\text{V}} \qquad \boldsymbol{c}_{\mathbf{t}}^{\text{V}} \in \mathbb{R}^{c} \end{aligned}$$

Name	Modality		Decoder State	
SS	Shared	$\boldsymbol{W}_e^{\mathrm{L}} = \boldsymbol{W}_e^{\mathrm{V}}$, $\boldsymbol{w}_a^{\mathrm{L}} = \boldsymbol{w}_a^{\mathrm{V}}$	Shared	$\boldsymbol{W}_q^{\mathrm{L}} = \boldsymbol{W}_q^{\mathrm{V}}$
SD			Dedicated	$\boldsymbol{W}_q^{\mathrm{L}} \neq \boldsymbol{W}_q^{\mathrm{V}}$
DS	Dedicated	$\boldsymbol{W}_e^{\mathrm{L}} \neq \boldsymbol{W}_e^{\mathrm{V}}$, $\boldsymbol{w}_a^{\mathrm{L}} \neq \boldsymbol{w}_a^{\mathrm{V}}$	Shared	$\boldsymbol{W}_q^{\mathrm{L}} = \boldsymbol{W}_q^{\mathrm{V}}$
DD			Dedicated	$\boldsymbol{W}_q^{\mathrm{L}} \neq \boldsymbol{W}_q^{\mathrm{V}}$

Table 7.2: Sharing strategies for multimodal attention: The name consists of the initials of "Shared" and "Dedicated" for modality and decoder state projections, respectively.

7.2.1 Feature Normalization

The spatial features are extracted after a ReLU convolutional layer that rectifies its input into $[0, \infty]$. On the other hand, the non-linearities in GRUs and in our baseline NMT in general are based on the tanh activation which squeezes its input to $[-1, 1]$. Our initial attempts to attentive MMT models in 2016 (Caglayan et al., 2016a) and 2017 (Caglayan et al., 2017a) editions of the shared task, had significantly poor performance compared to our respective baselines. We hypothesize that the reason behind this may be the activation ranges of language and visual features in the network which hinders the learning dynamics. Specifically, the unbounded visual features may easily saturate the tanh neurons in the network unless special care has been taken to adjust the random initialization scheme of the network weights. In Caglayan et al. (2018), we take a simpler normalization approach by following previous empirical evidence in VQA research (Kazemi and Elqursh, 2017; Yu et al., 2017) showing the benefit of applying L_2 normalization over the channel dimension of spatial features. Specifically, this ensures that the L_2 norm of each of the 64 (8x8) spatial feature vectors ($\in \mathbb{R}^{2048}$) is 1. The normalization step comes right after the extraction of the spatial features $\mathbf{F}$ (equation 7.4).

7.3 Sharing Strategies

In order to understand the effect of sharing the attention across the modalities, we propose four different strategies (Caglayan et al., 2016b) that are summarized in Table 7.2. When a parameter is shared, it is reused in both attention layers $\{\mathsf{ATT}^{\mathrm{L}}, \mathsf{ATT}^{\mathrm{V}}\}$ enforcing the model to learn a shared representation – for the outcome of that transformation – to minimize the training loss. When parameters are dedicated to modalities, the model would have more flexibility to independently optimize the corresponding transformation parameters. Unlike Firat et al. (2017) where a single attention is shared across multiple languages, we believe that a dedicated visual attention may be more appropriate

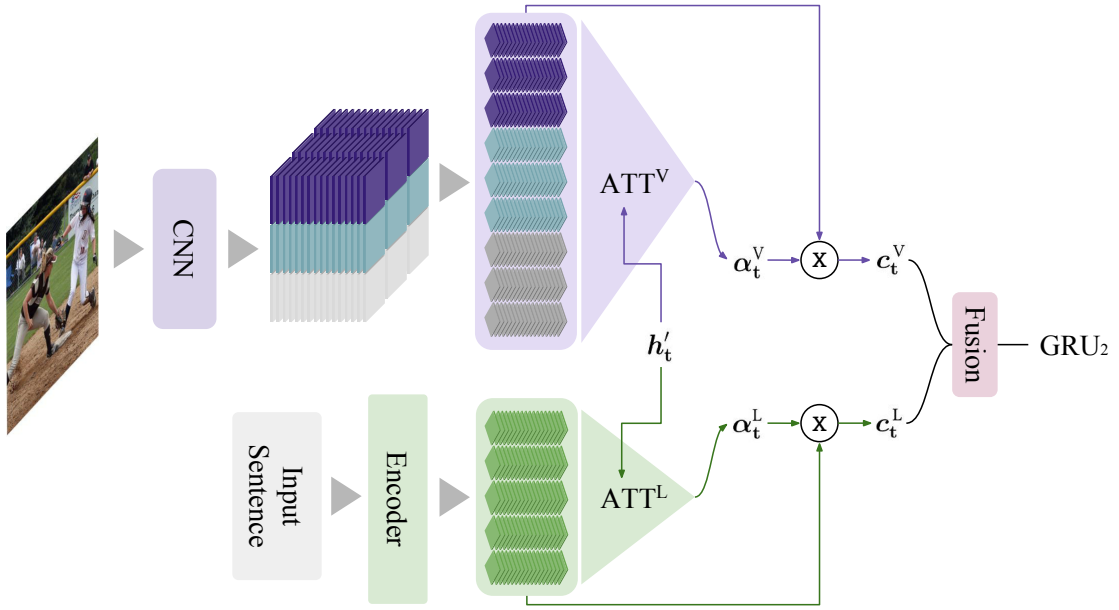


Figure 7.2: NMT with multimodal attention mechanism: modality specific contexts reach the fusion module which aims to compress the representations into a single vector.

for MMT simply because of the radically different nature of the modalities *i.e.* jointly learned embeddings and encoder for the language and visual features transferred from an image classification task.

7.4 Multimodal Fusion

The multimodal attention mechanism computes two modality specific context vectors $\{\boldsymbol{c}_t^{\mathrm{L}}, \boldsymbol{c}_t^{\mathrm{V}}\}$, independent from the choice of sharing strategy. A fusion has to be performed to compress these contexts into a single vector which would then be used as the input to the GRU$_2$ layer (Figure 7.2). A linear transformation already exists at this step for the textual NMT architecture (equation 7.2) to project from c-dimensional context space to the d-dimensional decoder input $\boldsymbol{i}_t$ (equation 7.3). The multimodal fusion is thus an extension to that step that considers both contexts when performing the projection. The following defines the SUM and CONCAT fusion methods proposed in Caglayan et al. (2016b):

$$\boldsymbol{i}_t = \phi\left(\boldsymbol{W}_d\ \left(\boldsymbol{c}_t^{\mathrm{L}} + \boldsymbol{c}_t^{\mathrm{V}}\right)\right) \quad = \phi\left(\boldsymbol{W}_d\, \boldsymbol{c}_t^{\mathrm{L}} + \boldsymbol{W}_d\, \boldsymbol{c}_t^{\mathrm{V}}\right) \qquad \text{SUM FUSION}$$

$$\boldsymbol{i}_t = \phi([\boldsymbol{W}_d^{\mathrm{L}}; \boldsymbol{W}_d^{\mathrm{V}}] \begin{bmatrix} \boldsymbol{c}_t^{\mathrm{L}} \\ \boldsymbol{c}_t^{\mathrm{V}} \end{bmatrix}) = \phi\left(\boldsymbol{W}_d^{\mathrm{L}}\, \boldsymbol{c}_t^{\mathrm{L}} + \boldsymbol{W}_d^{\mathrm{V}}\, \boldsymbol{c}_t^{\mathrm{V}}\right) \qquad \text{CONCAT FUSION}$$

The difference between CONCAT and SUM is that the former uses dedicated parameters $\{\boldsymbol{W}_d^{\mathrm{L}}, \boldsymbol{W}_d^{\mathrm{V}}\}$ while the latter shares a single $\boldsymbol{W}_d$ for the context transformations. In fact,

EN→DE	METEOR		BLEU	
System	SUM	CONCAT	SUM	CONCAT
NMT	58.4 ± 0.3		38.9 ± 0.8	
DD	56.3 ± 0.4 (↓ 2.1)	56.9 ± 0.3 (↓ 1.5)	36.5 ± 0.6 (↓ 2.4)	37.4 ± 0.6 (↓ 1.5)
+ L_2	58.2 ± 0.3 (↓ 0.2)	58.0 ± 0.2 (↓ 0.4)	39.3 ± 0.4 (↑ 0.4)	38.6 ± 0.3 (↓ 0.3)
SS	56.4 ± 0.4 (↓ 2.0)	57.4 ± 0.1 (↓ 1.0)	37.0 ± 0.4 (↓ 1.9)	37.1 ± 0.4 (↓ 1.8)
+ L_2	58.2 ± 0.3 (↓ 0.2)	58.4 ± 0.2	38.7 ± 0.2 (↓ 0.2)	38.9 ± 0.2
SD	56.9 ± 0.1 (↓ 1.5)	57.0 ± 0.3 (↓ 1.4)	37.4 ± 0.6 (↓ 1.5)	37.3 ± 0.3 (↓ 1.6)
+ L_2	58.4 ± 0.7	58.1 ± 0.2 (↓ 0.3)	39.0 ± 0.6 (↑ 0.1)	38.6 ± 0.7 (↓ 0.3)
DS	56.8 ± 0.4 (↓ 1.6)	57.2 ± 0.2 (↓ 1.2)	37.3 ± 0.6 (↓ 1.6)	36.8 ± 0.4 (↓ 2.1)
+ L_2	58.7 ± 0.1 (↑ 0.3)	58.5 ± 0.1 (↑ 0.1)	39.4 ± 0.1 (↑ 0.5)	39.2 ± 0.4 (↑ 0.3)

Table 7.3: The impact of L_2 normalization on MMT performance on test2016: All differences are against the baseline NMT.

The SS model with SUM fusion (SS-SUM) implements a "completely shared" multimodal attention while the DD model with CONCAT fusion (DD-CAT) performs a "completely dedicated" multimodal attention with the least amount of crossmodal interaction involved. Another popular fusion method is the hierarchical attention (Libovický and Helcl, 2017) that employ a third attention mechanism on top of the multimodal contexts.

7.5 Results & Analysis

We first evaluate the performance of eight MMT variants that result from combining four sharing strategies with two fusion methods. We start by comparing BLEU and METEOR scores of these systems with and without L_2 normalization on the test2016 set of English→German direction (Table 7.3). A quick look at the results reveal that without normalization, the results are far from being competitive, achieving 1.5 METEOR and 1.8 BLEU less than the baseline on average. With feature normalization however, the systems reach the baseline performance, with DS models even slightly improving over it (0.3 METEOR and 0.5 BLEU points for DS-SUM). In order to understand the qualitative impact of normalization, we visualize the language and visual attentions for one of the MMT variants (SS-SUM) in Figure 7.3. The example shows that with normalized features, the model produces a meaningful spatial attention where it first focuses on the "person" and then highlights the "mountain" in the background.

We now extend the results to English→French and report metrics on both test sets (Table 7.4). First of all, we observe that all **German** MMT models perform at least as

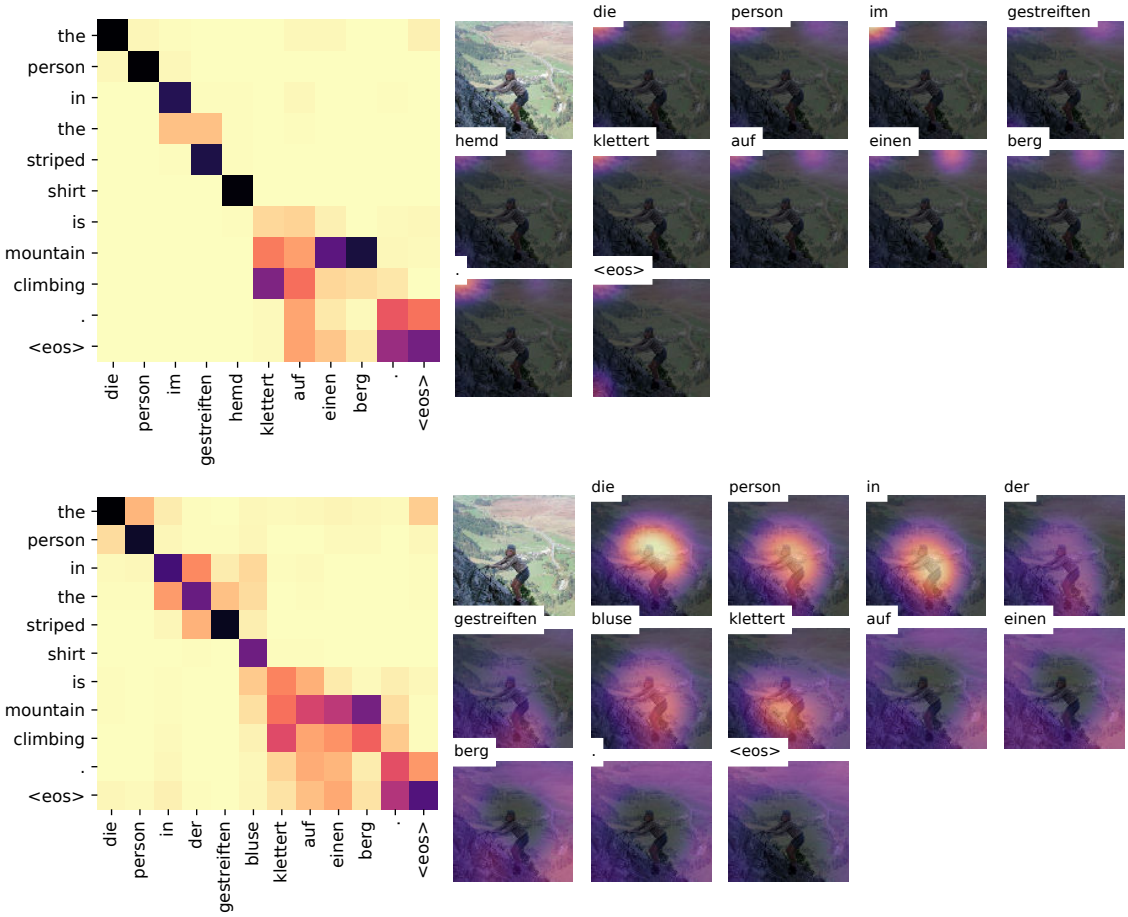


Figure 7.3: The qualitative impact of L_2 normalization for multimodal attention: (Top, unnormalized) visual attention does not make sense (bottom, normalized) the attention shifts focus from "person" to the "mountain". The model is SS-SUM.

good as the baseline on test2017 although the differences are only significant for the DS systems (p-value ≤ 0.05 according to multeval (Clark et al., 2011)). On average, the DS-CAT performs better than the baseline with 0.6 and 0.4 BLEU and METEOR points, respectively. The results are less promising for **French** where none of the systems are significantly different on any test set. Interestingly, we observe the same saturating behavior as in the case of simple MMT (section 6.2, p. 65): French MMT systems are quite stable and barely move in terms of automatic metrics while all German MMTs perform at least as good as the baseline on test2017. Although it is not possible to draw a conclusion about the individual strengths of the models, we notice that the top ranked MMTs for both languages have modality dedicated attentions *i.e.* DS-CAT for German and DD-CAT for French. The choice of multimodal fusion does not seem to make a crucial difference. In the following, we briefly look at the results of sentence level analysis protocol introduced in the previous chapter (section 6.2.1, p. 66).

	test2016		test2017	
	BLEU	METEOR	BLEU	METEOR
		English→German		
NMT	38.9 ± 0.8	58.4 ± 0.3	32.1 ± 1.1	52.5 ± 0.7
SD-SUM	39.0 ± 0.6 (↑ 0.1)	58.4 ± 0.7	32.4 ± 0.6 (↑ 0.3)	52.5 ± 0.2
DD-SUM	39.3 ± 0.4 (↑ 0.4)	58.2 ± 0.3 (↓ 0.2)	32.5 ± 1.1 (↑ 0.4)	52.6 ± 0.4 (↑ 0.1)
DD-CAT	38.6 ± 0.3 (↓ 0.3)	58.0 ± 0.2 (↓ 0.4)	32.3 ± 0.8 (↑ 0.2)	52.6 ± 0.5 (↑ 0.1)
SS-CAT	38.9 ± 0.2	58.4 ± 0.2	32.3 ± 0.2 (↑ 0.2)	52.6 ± 0.4 (↑ 0.1)
SS-SUM	38.7 ± 0.2 (↓ 0.2)	58.2 ± 0.3 (↓ 0.2)	32.7 ± 0.4 (↑ 0.6)	52.7 ± 0.2 (↑ 0.2)
SD-CAT	38.6 ± 0.7 (↓ 0.3)	58.1 ± 0.2 (↓ 0.3)	32.5 ± 0.2 (↑ 0.4)	52.8 ± 0.2 (↑ 0.3)
DS-SUM	39.4 ± 0.1 (↑ 0.5)	58.7 ± 0.1 (↑ 0.3)	32.6 ± 0.4 (↑ 0.5)	52.9 ± 0.3 (↑ 0.4)
DS-CAT	39.2 ± 0.4 (↑ 0.3)	58.5 ± 0.1 (↑ 0.1)	32.7 ± 0.2 (↑ 0.6)	52.9 ± 0.5 (↑ 0.4)
		English→French		
NMT	61.4 ± 0.3	76.4 ± 0.2	54.4 ± 0.3	71.1 ± 0.2
SD-CAT	61.0 ± 0.7 (↓ 0.4)	76.2 ± 0.4 (↓ 0.2)	53.8 ± 0.5 (↓ 0.6)	70.8 ± 0.4 (↓ 0.3)
DS-CAT	61.5 ± 0.1 (↑ 0.1)	76.3 ± 0.1 (↓ 0.1)	54.0 ± 0.1 (↓ 0.4)	70.8 ± 0.1 (↓ 0.3)
SS-CAT	61.3 ± 0.4 (↓ 0.1)	76.2 ± 0.3 (↓ 0.2)	54.2 ± 0.3 (↓ 0.2)	70.9 ± 0.4 (↓ 0.2)
DS-SUM	60.7 ± 0.0 (↓ 0.7)	76.0 ± 0.2 (↓ 0.4)	54.2 ± 0.4 (↓ 0.2)	71.0 ± 0.1 (↓ 0.1)
SS-SUM	61.2 ± 0.4 (↓ 0.2)	76.2 ± 0.2 (↓ 0.2)	54.0 ± 0.4 (↓ 0.4)	71.0 ± 0.1 (↓ 0.1)
DD-SUM	61.6 ± 0.4 (↑ 0.2)	76.5 ± 0.4 (↑ 0.1)	54.2 ± 0.3 (↓ 0.2)	71.0 ± 0.2 (↓ 0.1)
SD-SUM	61.3 ± 0.2 (↓ 0.1)	76.2 ± 0.0 (↓ 0.2)	54.3 ± 0.2 (↓ 0.1)	71.1 ± 0.2
DD-CAT	61.2 ± 0.6 (↓ 0.2)	76.3 ± 0.3 (↓ 0.1)	54.1 ± 0.7 (↓ 0.3)	71.2 ± 0.1 (↑ 0.1)

Table 7.4: Combined results on test2016 and test2017: Highlighted scores are significantly different than the NMT (p-value ≤ 0.05). Results ordered by test2017 METEOR.

7.5.1 Sentence Level Analysis

Figure 7.4 plots the percentages of ties, wins and losses on test2017 for both language pairs. The conclusions are pretty coherent with the SMMT analysis (section 6.2.1, p. 66): On average, 28.9% and 15.5% of French and German multimodal translations preserve their METEOR consistently across AMMT variants with a standard deviation of 1.2%. Let us remind that these averages are once again almost the same as the retrained "control" and thus not related at all to multimodality. The German DS-CAT improves %45.6 of the translations while deteriorating on 39.1% (6.5% "wins - losses" gap) whereas the same system for French is the worse in this aspect with −5.7% "wins - losses" gap. In overall, the fact that the sentence level breakdowns for SMMTs and AMMTs look pretty similar hints at the fact that the behavior of the models is mostly driven by the language signal as well as the test set characteristics rather than the type of multimodality introduced. In other words, the models do not seem to be stimulated by the visual input.

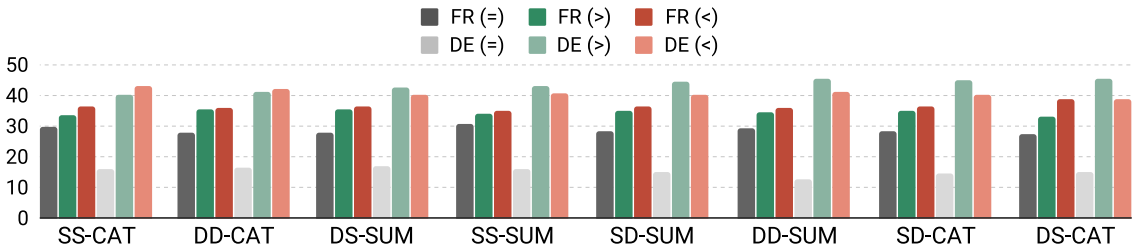


Figure 7.4: Sentence level METEOR breakdown for attentive MMT systems: The results are ordered by German (DE) "wins - losses" gap.

7.5.2 Analysis of the Visual Attention

We previously saw in Figure 7.3 (bottom) that the language attention is able to preserve its certainty (peakiness) throughout the translation decoding despite the fact that the `SS-SUM` model has a completely shared attention across both modalities. To better understand the behaviour of different sharing strategies as well as the type of multimodal fusion, we collect statistics during the decoding of `test2016` sentences and compute normalized entropies for language and visual attention mechanisms. The normalization is performed by dividing the entropy per sample by the uniform entropy, taking into account the number of source words for each decoded sentence in the case of language attention. For example, a visual attention that "always" assigns a probability of $\frac{1}{8\times8}$ to each position in the 8x8 convolutional feature maps, obtains a normalized entropy of 100%, indicating the highest uncertainty. The final entropy is computed by simply taking the average of per sample entropies. Figure 7.5 plots the computed entropies across the explored AMMT variants. First of all, we can see that the uncertainty of the language attention does not seem to be affected by the multimodality and behaves similarly to the baseline `NMT`. In contrast, the uncertainty of the visual attention consistently increases as the attention becomes more and more shared across modalities. In fact, the visual attention of `SS-SUM` turns out to be "almost" uniform.

Finally, we visualize the spatial attention of the models on a specific example of `test2016` in Figure 7.6. Since the entropy of each model radically differs from each other, it is impossible to visualize the heatmaps with a normalized scale *i.e.* the magnitudes of the attention are not quite comparable across models. Nevertheless, the plot still gives an idea about the internal view of each model: Although quite uniform, the `SS-SUM` produces a plausible attention where tiny differences in the probability mass determine the focus. On the other hand, the peakiness of the visual attention increases as the multimodal attention becomes more and more dedicated.

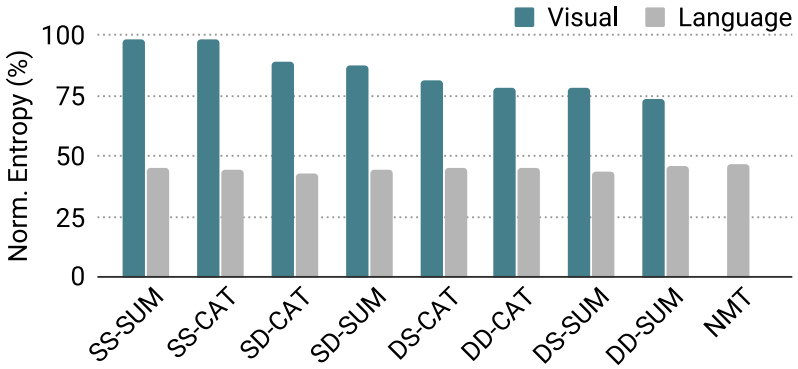


Figure 7.5: The normalized entropies of attention distributions: the language attention has consistently lower entropy than the visual attention which converges to uniform distribution when the attention is completely shared.

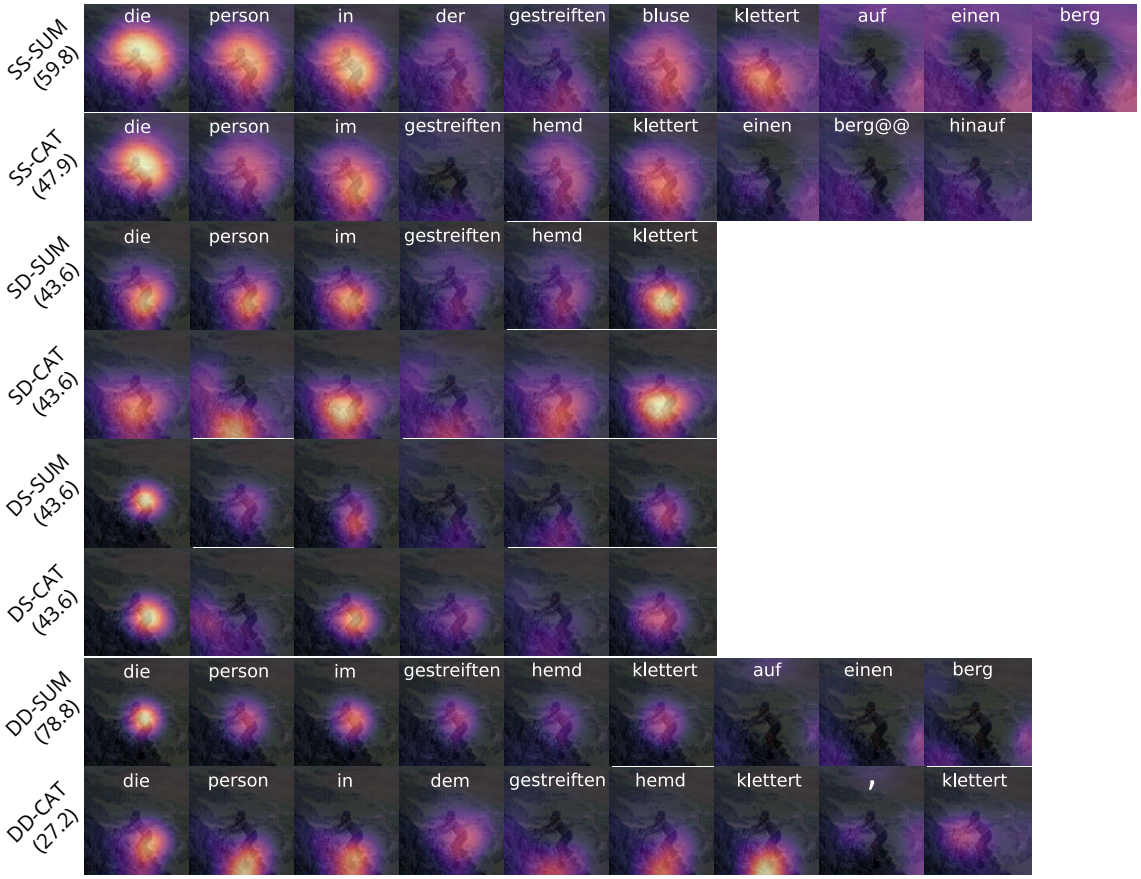


Figure 7.6: Comparison of visual attention across MMT variants: the attention becomes more peaky (less uniform) when going from shared to dedicated variants. The corresponding sentence level METEOR for each model is given inside parentheses. The reference translation is *"die person im gestreiften shirt klettert auf einen berg (the person in the striped shirt climbs on a mountain)"*. The four systems in the middle miss "on a mountain" part. `DD-SUM` correctly translates the sentence but prefers "hemd" over "shirt", which is penalized by METEOR.

	EN→DE (test2017)	
	BLEU	METEOR
NMT	32.1 ± 1.1	52.5 ± 0.7
DD	32.3 ± 0.8 (↑ 0.2)	52.6 ± 0.5 (↑ 0.1)
SS	32.3 ± 0.2 (↑ 0.2)	52.6 ± 0.4 (↑ 0.1)
SD	32.5 ± 0.2 (↑ 0.4)	52.8 ± 0.2 (↑ 0.3)
UVA	33.0 ± 0.3 (↑ 0.9)	52.8 ± 0.1 (↑ 0.3)
DS	32.7 ± 0.2 (↑ 0.6)	52.9 ± 0.5 (↑ 0.4)

Table 7.5: Uniform visual attention (UVA) on German test2017: UVA obtains the best average BLEU as well as a competitive METEOR. Highlighted scores are significantly different than the NMT (p-value ≤ 0.05). All systems are CONCAT variants.

7.6 Uniform Visual Attention

Although the entropy analysis suggests that dedicating the attention mechanism decreases the uncertainty of the spatial focus, the automatic evaluation metrics do not reflect any preference towards a specific kind of attention. Moreover, the almost uniform shared attention variants seem to produce quite plausible attention maps (Figure 7.6). This raises an interesting question: How important is the spatial certainty of the visual attention for the model performance? In order to answer this, we propose an ablation experiment which consists of replacing the learnable visual attention layer ATT^{V} with a dummy layer that explicitly assigns a uniform probability of $\frac{1}{8\times 8}$ to each spatial position. The final model that we call "uniform visual attention (UVA)" still uses the spatial features to compute the visual context $\boldsymbol{c}_t^{\text{V}}$ but this context no longer depends on the hidden state $\boldsymbol{h}'_t$ of the decoder *i.e.* it stays constant across the decoding steps. In fact, this amounts to using the global visual features $\boldsymbol{f}$ at each decoding step since $\boldsymbol{f}$ is the global average pooled version of the spatial features (section 2.6.2, p. 26).

Table 7.5 compares the BLEU and METEOR scores of AMMT systems to UVA with the multimodal fusion operation set to "concatenation" for each model. We see that the UVA model obtains the highest average BLEU score and is significantly different than the baseline according to multeval. It also outperforms the "almost uniform" SS variant on average metrics. These results suggest that the model benefits more from a constant and spatially unaware visual signal compared to a noisy version of it which evolves throughout the decoding steps.

	System	BLEU	METEOR	Description
Caglayan et al. (2016a)	RNN	29.3 (↓ 4.6)	48.5 (↓ 4.3)	Shared Attention
Helcl and Libovický (2017)	RNN	31.9 (↓ 2.7)	49.4 (↓ 2.3)	Hierarchical Attention
Calixto et al. (2016)	RNN	28.8	49.6	Separate Attention
Arslan et al. (2018)	TF	41.0 (↑ 2.4)	53.5 (↓ 1.5)	Parallel Attention
Calixto et al. (2017b)	RNN	36.5 (↑ 2.8)	55.0 (↑ 2.7)	β-gated Attention
Caglayan et al. (2017a)	RNN	37.0 (↓ 1.1)	57.0 (↓ 0.3)	Separate Attention
Libovický et al. (2018)	TF	38.6 (↑ 0.3)	57.4 (↑ 0.7)	Parallel Attention
Caglayan et al. (2016a)	PBMT	36.2 (0.0)	57.5 (↑ 0.1)	Reranking (Visual NLM)
Delbrouck and Dupont (2017a)	RNN	40.5	57.9	BNM + Enc. Attention
SMMT (chapter 6)	RNN	39.0 (↑ 0.1)	58.5 (↑ 0.1)	`EDINIT`
		39.5 (↑ 0.6)	58.6 (↑ 0.2)	`DINIT`
AMMT (This chapter)	RNN	39.4 (↑ 0.5)	58.7 (↑ 0.3)	`DS-SUM` + L_2

Table 7.6: Comparison of state-of-the-art AMMTs on German `test2016`: TF stands for Transformer (Vaswani et al., 2017). We do not report ensemble results to ensure a fair comparison. The relative differences inside parentheses are with respect to the baseline MTs reported in those works. The results are sorted by METEOR.

7.7 Comparison to State-of-the-art

Table 7.6 compares our best attentive MMTs to a selection of state-of-the-art systems. I also include the SMMT systems from the previous chapter to provide a global view of all models presented in this thesis. The first conclusions are pretty much the same as SMMTs (section 6.3, p. 69): better baselines seem to benefit less from the visual modality. Different from SMMTs though, we observe that attentive systems struggle more to maintain the baseline performance. In fact, the only system that substantially improves over their baseline with respect to both metrics is Calixto et al. (2017b). In our case, this issue is now addressed with L_2 normalization which allows our models to at least perform as good as the baseline on average. In overall, both our SMMT and AMMT systems perform equally well and obtain state-of-the-art scores with respect to automatic evaluation metrics. The gains in BLEU are slightly higher than METEOR, however it should be noted that BLEU exhibits a higher variance – at least in the case of Multi30K – as shown in the detailed quantitative results.

7.8 Summary

In this chapter, I presented several attentive MMT systems with different sharing levels and multimodal fusion techniques. I first showed how L_2 normalization of spatial features is crucial for these models to reach the baseline performance, then conducted a quantitative analysis on English→German and English→French translation tasks of Multi30K. Although some of the models were shown to be significantly better than the baseline for German, we struggle to reach a global conclusion about the performance of the AMMT systems. After gaining some insights from the entropies of attention distributions, we conduct a contrastive experiment where the visual attention is replaced with a dummy layer which constantly puts a uniform attention over the image features. The fact that this model obtains competitive scores as well raises an obvious question about whether the quantitative gains can be solely attributed to multimodality or not.

We also observe that the quantitative results for AMMTs are mostly coherent with the SMMT models in the sense that both approaches yield mild improvements for German while barely moving for French. This is interesting as it strongly points out that it is the linguistic traits of the underlying language pairs and the dataset which seem to dominate the final performance trends of the models rather than the visual feature type or the interaction scheme explored. The next chapter attempts to tackle these concerns by providing a set of ablation experiments to probe the visual awareness of SMMT and AMMT models explored throughout the thesis.

CHAPTER 8

Deeper Analysis of MMT Systems

In previous chapters, we explored several multimodal integration methods for NMT by first using the global visual features (SMMT) and then moving on to more sophisticated attentive approaches (AMMT) which incorporate spatially aware features. Upon various quantitative analyses and manual inspection of the model dynamics, we find it hard to reach a conclusion on the strengths and weaknesses of the proposed architectures in terms of their ability to integrate the visual modality.

Recent evidence from the literature also suggest that the benefits of the current MMT approaches are little to none on Multi30K. Lala et al. (2018) show that when used to rerank a list of translation candidates, their multimodal WSD method is not any better than the monomodal counterpart. Elliott (2018) demonstrate that the performance of state-of-the-art MMTs is marginally influenced when they are adversarially attacked by incongruent images *i.e.* when source sentences are paired with images not being the ones described by those sentences. After experimenting with a plethora of visual features and integration methods, Grönroos et al. (2018) also find out that their English→French MMT is not negatively affected at all by the adversarial evaluation, corroborating the findings of Elliott (2018). Finally, the organizers of the shared task point out that "the integration of visual modality does not seem to help reliably" (Elliott et al., 2017) and there may be a need for a more challenging task & dataset for which the images would be indispensable (Barrault et al., 2018). We believe that the underlying reason behind these negative conclusions may be the simple, short and repetitive nature of the Multi30K dataset rendering the source sentences sufficient for the translation task. In turn, this may prevent the visual modality from intervening in the learning process if the model sees no benefit from it when minimizing the loss.

To investigate our hypothesis, here we propose to systematically deprive the models from textual context primarily by masking out visually depictable words from the source sentences (Caglayan et al., 2019a). We then evaluate these new models using the

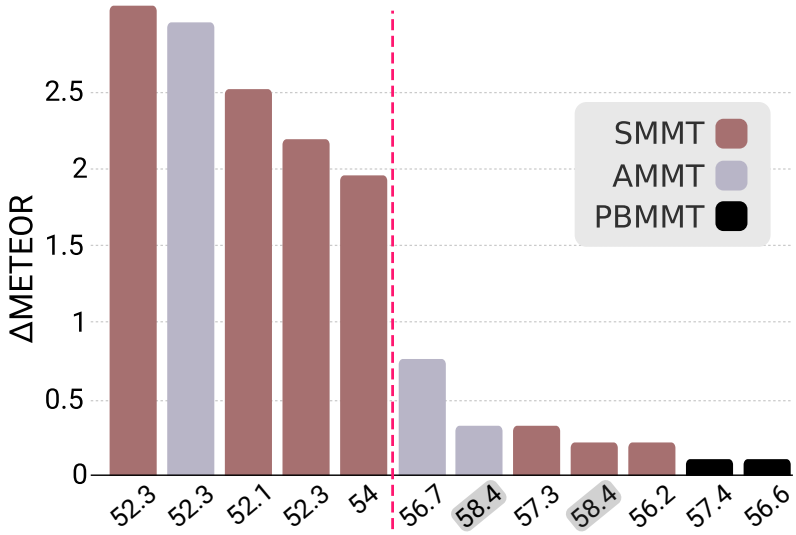


Figure 8.1: State-of-the-art multimodal gains over corresponding baselines: The x-axis shows the baseline METEOR scores on English→German `test2016` for a set of state-of-the-art systems. The systems with shaded scores are the best SMMT and AMMT systems from this thesis.

adversarial protocol in order to assess their visual sensitivity. But before doing so, we revisit the state-of-the-art English→German models (section 4.2.4, p. 55) once again to discuss the nature of the previously demonstrated multimodal gains.

Figure 8.1 shows the METEOR gains relative to the baseline MTs reported in the corresponding papers. We make sure that neural MMT systems are compared to NMT baselines while multimodal PBMTs (PBMMT) are compared to PBMT baselines. The plot makes it clear that the improvements due to the visual modality are only prominent if the underlying baselines are not optimal *i.e.* they are not able to fully exploit the language signal for some reason. The dashed vertical line sets an hypothetical boundary after which all baselines obtain a METEOR ≥ 54 and all corresponding multimodal gains are ≤ 0.7 METEOR. Recall that the multimodal gains for our French MMT17 systems no longer hold for the retrained systems where the new baseline is significantly better than the old one (section 6.2.2, p. 68). We thus posit that the retrained French systems crossed a similar hypothetical boundary after which the benefit of the visual modality becomes little to none. This brings us to the previously introduced question of "whether the quantitative gains can be solely attributed to multimodality or not". In order to answer this question at least for our current SMMT and AMMT systems, we now describe the "adversarial evaluation" method (Elliott, 2018).

	BLEU			METEOR		
	Congruent	Incongruent	Δ	Congruent	Incongruent	Δ
NMT	38.9 ± 0.8			58.4 ± 0.3		
VBOS	38.9 ± 0.1	39.0 ± 0.1	↑0.1	58.3 ± 0.2	58.4 ± 0.1	↑0.1
TMUL	38.8 ± 0.1	38.9 ± 0.1	↑0.1	58.3 ± 0.2	58.3 ± 0.1	0.0
SMUL	39.0 ± 0.6	39.0 ± 0.6	0.0	58.2 ± 0.4	58.2 ± 0.3	0.0
EINIT	39.6 ± 0.4	39.5 ± 0.7	↓0.1	58.4 ± 0.2	58.5 ± 0.3	↑0.1
DINIT	39.5 ± 0.1	39.4 ± 0.1	↓0.1	58.6 ± 0.3	58.6 ± 0.3	0.0
EDINIT	39.0 ± 0.4	38.8 ± 0.4	↓0.2	58.5 ± 0.3	58.4 ± 0.4	↓0.1
EMUL	38.6 ± 0.4	38.2 ± 0.1	↓0.4	58.1 ± 0.3	57.8 ± 0.4	↓0.3

Table 8.1: Adversarial evaluation of SMMT systems on English→German `test2016`: The incongruently decoded `EMUL` system is significantly different (p-value ≤ 0.05) than its congruent counterpart with respect to METEOR. The Δ's are computed by subtracting the congruent mean from the incongruent mean.

8.1 Adversarial Evaluation

The protocol starts with decoding a given test set using incongruent visual features. The incongruence is achieved by shuffling the order of the visual features so that a source sentence $\mathbf{X}_i$ is explicitly aligned to a "wrong" visual feature $\mathbf{V}_{j \neq i}$. Consequently, an MMT system capable of integrating the visual modality would likely deteriorate in terms of automatic evaluation metrics. For a given test set, Elliott (2018) repeat the decoding process five times by re-shuffling the order each time, in order to filter out noisy measurements that can be caused by a specific shuffle. Here we take a slightly different approach and we create a single incongruent test set with reversed feature order *i.e.* the source sentences $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ are deliberately misaligned to visual features $\{\mathbf{V}_N, \mathbf{V}_{N-1}, \ldots, \mathbf{V}_1\}$ where N denotes the size of the test set. Finally, we decode this incongruent test set for each of the three runs of a given model and compute the mean and the standard deviation of BLEU and METEOR using `multeval` tool (Clark et al., 2011) as in the previous chapters. This way, we are able to leverage the statistical significance tests of `multeval` to evaluate an incongruently decoded model to its congruently decoded baseline.

Table 8.1 shows the results for English→German SMMT systems. First of all, we notice that the averaged metric shifts (Δ) due to incongruent decoding are quite small, with the `EMUL` system deteriorating the most. `EMUL` is also the only system for which the incongruently decoded translations are significantly different than the congruent counterparts with respect to METEOR. Other than the `EMUL` and `EDINIT` variants, the rest are barely reacting to the misaligned visual features, some of them even showing slight improvements, an interesting effect also observed by Grönroos et al. (2018).

	BLEU			METEOR		
	Congruent	Incongruent	Δ	Congruent	Incongruent	Δ
NMT	38.9 ± 0.8			58.4 ± 0.3		
SS-SUM	38.7 ± 0.2	39.3 ± 0.3	↑0.6	58.2 ± 0.3	58.5 ± 0.1	↑0.3
DD-CAT	38.6 ± 0.3	38.6 ± 0.2	0.0	58.0 ± 0.2	58.0 ± 0.3	0.0
UVA	39.3 ± 0.4	39.2 ± 0.6	↓0.1	58.2 ± 0.3	58.3 ± 0.4	↑0.1
SD-SUM	39.0 ± 0.6	38.9 ± 0.8	↓0.1	58.4 ± 0.7	58.4 ± 0.6	0.0
SD-CAT	38.6 ± 0.7	38.5 ± 0.6	↓0.1	58.1 ± 0.2	58.1 ± 0.3	0.0
DD-SUM	39.3 ± 0.4	39.0 ± 0.7	↓0.3	58.2 ± 0.3	58.3 ± 0.2	↑0.1
DS-SUM	39.4 ± 0.1	39.1 ± 0.2	↓0.3	58.7 ± 0.1	58.6 ± 0.2	↓0.1
SS-CAT	38.9 ± 0.2	38.6 ± 0.4	↓0.3	58.4 ± 0.2	58.2 ± 0.3	↓0.2
DS-CAT	39.2 ± 0.4	38.8 ± 0.4	↓0.4	58.5 ± 0.1	58.2 ± 0.1	↓0.3

Table 8.2: Adversarial evaluation of AMMT systems on English→German test2016: The incongruently decoded DS-CAT system is significantly different (p-value ≤ 0.05) than its congruent counterpart with respect to BLEU.

We observe a similar behavior among the AMMT systems (Table 8.2) although they seem to deteriorate slightly more than the SMMT systems. The dedicated attention variant DS-CAT significantly worsens by incongruent decoding with respect to BLEU. The completely shared variant SS-SUM exhibits nonnegligible average improvements when decoded incongruently, a phenomenon which strongly suggests that the visual modality behaves as a structured noise which substantially influences the output probability distribution at translation decoding time – at least – for this model.

Globally, the adversarial evaluation results for both types of MMT suggest one clear thing: The visual signal is not a vital contributor to the multimodal reasoning ability as none of the models completely breaks apart when challenged with unrelated visual features. In other words, the modalities are far from being cooperative. In theory, this should not reject the (weak) possibility that the visual modality may be providing a complementary signal for the models that consistently suffer from incongruence. However, a manual inspection of the translations for these systems reveal no systematic signs for that: For the **incongruent** DS-CAT system, a sentence that substantially deteriorates actually replaces the word "footballspieler" with its hyphenated version "football-spieler" whereas another one reaches 100% METEOR by adding the previously missing "in der stadt (in the city)" phrase to its translation. The first example also shows how fragile the automatic evaluation is when performed with a single set of references.

	Description	Source Sentence							Image
$\mathcal{D}$	Original	a	lady	in	a	blue	dress	singing	
$\mathcal{D}_N$	Entity Masking	a	[v]	in	a	blue	[v]	singing	
$\mathcal{D}_4$	Prog. Masking (k=4)	a	lady	in	a	[v]	[v]	[v]	
$\mathcal{D}_2$	Prog. Masking (k=2)	a	lady	[v]	[v]	[v]	[v]	[v]	
$\mathcal{D}_0$	Prog. Masking (k=0)	[v]	[v]	[v]	[v]	[v]	[v]	[v]	

Table 8.3: A depiction of the proposed text degradations: $\mathcal{D}$ is the original test set.

8.2 Degradation Methods

In this section we propose to explicitly degrade the *source sentences* in Multi30K training and test sets at different scales. The idea here is to understand whether the explored models can gain multimodal reasoning abilities by learning to refer to the images when the information no longer exists in the source sentence. In the following, we describe two approaches, namely, the progressive and entity masking (Table 8.3), and then proceed with quantitative and qualitative analyses.

8.2.1 Progressive Masking

A progressively masked variant $\mathcal{D}_k$ replaces all but the first k tokens of source sentences with `[v]`. These tokens are further considered as OOVs during training and test time. Overall, we form 16 degraded variants $\mathcal{D}_k$ (Table 8.3) where $k \in \{0, 2, \ldots, 30\}$. We stop at $\mathcal{D}_{30}$ since 99.8% of the sentences in Multi30K are shorter than 30 words. $\mathcal{D}_0$ is a special case where the only information that the models can extract from a source sentence is its length. This is interesting as an NMT model trained on $\mathcal{D}_0$ will only be able to generate a single sentence per source sentence length, since all sentences with the same number of words look the same to the decoder. On the other hand, an MMT has the potential to remedy that problem as it also has access to an auxiliary source of information, namely, the image features. In turn, the MMT system will behave as an image captioning system which can also guess the number of target words to be generated.

Progressive masking does not guarantee systematic removal of visual context, but simulates an increasingly low-resource scenario where the models have only access to sentence prefixes. The NMT and MMT models trained on a progressively masked variant no longer perform machine translation but translation completion. Although this may sound unrealistic, the task is still interesting as an NMT model will purely reflect the intrinsic biases of the dataset while MMT models will potentially apply debiasing with the help of the visual modality.

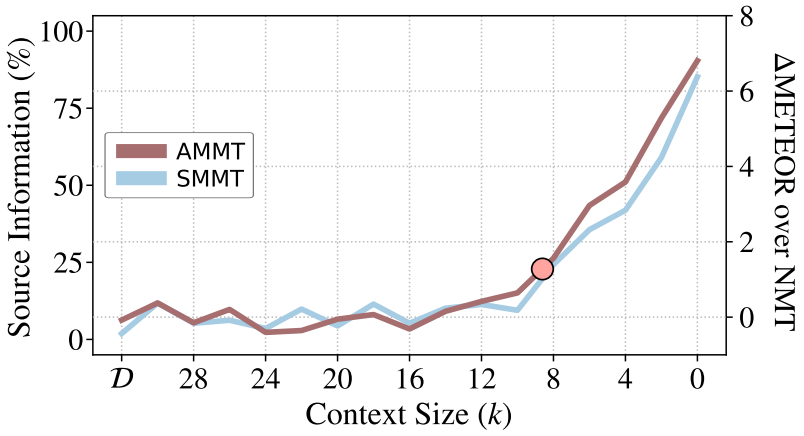


Figure 8.2: Multimodal gain in METEOR for progressive masking: The dashed gray curve indicates the percentage of non-masked words in the training set. The large dot marks the point after which the gap surpasses 1 METEOR.

Results

To evaluate the models for progressive masking, we pick the completely dedicated AMMT (`DD-CAT`) and the encoder-decoder initialization SMMT (`EDINIT`) as our target models. For each progressively masked dataset $\mathcal{D}_k$, we train the two MMTs along with the baseline NMT on English→French task. After the training, we follow the usual pipeline for decoding and evaluating the models. We compute the gain in METEOR over the **masked** NMT by averaging the gains of each model across the three runs.

Figure 8.2 shows the evolution of the multimodal gain as the sentence prefixes get shorter and shorter. The dashed gray curve marks the percentage of non-masked words in the training set *i.e.* the amount of remaining information. We observe that the improvements become prominent ($\geq$ 1 METEOR) when the context size shrinks to $\sim$9 words which is equivalent to $\sim$68% source information ($\sim$32% information dropped). This point more or less reflects the average number of words per sentence which is $\sim$13 for the training set and $\sim$11 for the test set. After that point, the gap widens significantly, reaching $\sim$7 METEOR at $\mathcal{D}_0$. Finally, the SMMT consistently lags behind the AMMT by around 1 METEOR, showing for the first time that the spatial feature based MMTs are able to leverage more visual context than the global feature based ones.

Table 8.4 provides qualitative examples from a couple of progressive masking experiments. We can see that the AMMT system is able to produce surprisingly good sentences that reflect more than one aspect of the image. We have also checked to what extent the correctly predicted phrases co-occur within a same context in the training set. For the second example, “dansent dans une rue” occurs only once in the training set and it is not followed by “en ville”. For the third example, “maillot de bain rose” occurs in six sentences but none of them starts with “une femme”.

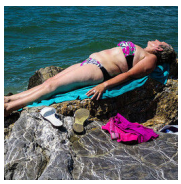
	SRC: trees are in front [v][v][v][v][v] REF: des **arbres** sont devant une grande **montagne** *(trees are in front of a big mountain)* NMT: des vélos sont devant un bâtiment en plein air *(bicycles are in front of an outdoor building)* AMMT: des **arbres** sont devant la **montagne** *(trees are in front of the mountain)* INC: des taxis sont devant la fenêtre d'une voiture *(taxis are in front of the window of a car)*
	SRC: girls wave purple flags [v][v][v][v][v][v][v] REF: des filles agitent des drapeaux violets tandis qu'elles défilent dans la rue *(girls wave purple flags as they parade down the street)* NMT: des filles en t-shirts violets sont assises sur des chaises dans une salle de classe *(girls in purple t-shirts are sitting on chairs in a classroom)* AMMT: des filles en costumes violets **dansent dans une rue en ville** *(girls in purple costumes dance on a city street)* INC: des filles en maillots rouges faisant du vélo dans une rue en ville *(girls in red shirts riding a bicycle in a city street)*
	SRC: an older woman in [v][v][v][v][v][v][v][v][v][v][v] REF: une femme âgée **en bikini** bronze sur **un rocher au bord de l'océan** *(an older woman in bikini is tanning on a rock at the edge of the ocean)* NMT: une femme âgée avec un t-shirt blanc et des lunettes de soleil est assise sur un banc *(an older woman with a white t-shirt and sunglasses is sitting on a bank)* AMMT: une femme âgée en **maillot de bain rose** est assise sur un **rocher au bord de l'eau** *(an older woman with a pink swimsuit is sitting on a rock at the seaside)* INC: une femme âgée en t-shirt blanc est debout à côté d'un grand arbre *(an older woman in white t-shirt is standing next to a large tree)*

Table 8.4: Progressive masking examples from English→French models: underlined and bold words highlight bad and **good** lexical choices, respectively. English translations are provided in parentheses. The red INC lines are incongruent AMMT outputs.

Finally, if we look at the incongruently decoded AMMT outputs, we can see that the models start to hallucinate, confirming that the effect of visual features is not random. In overall, we conclude that the models are able to guide the decoder to produce both fluent and visually adequate sentences and when doing so they are not merely retrieving sentences out of the training set. More examples are provided in appendix A. Table A.3 is especially interesting as it compares the successive outputs of the NMT and the MMT for a set of masked datasets $\mathcal{D}_k$.

8.2.2 Entity Masking

Here we take advantage from an extension of Flickr30K dataset which provides coreference chains to annotate *visually depictable* entities in the image descriptions (Plummer et al., 2015). Since Multi30K is derived from Flickr30K, we can exploit these annotations for the source-side `train`, `val` and `test2016` sentences. Specifically, we replace every annotated noun with a special token `[v]` as in the case of progressive masking. The annotations are not limited to single nouns but can extend to noun phrases such as "a blue

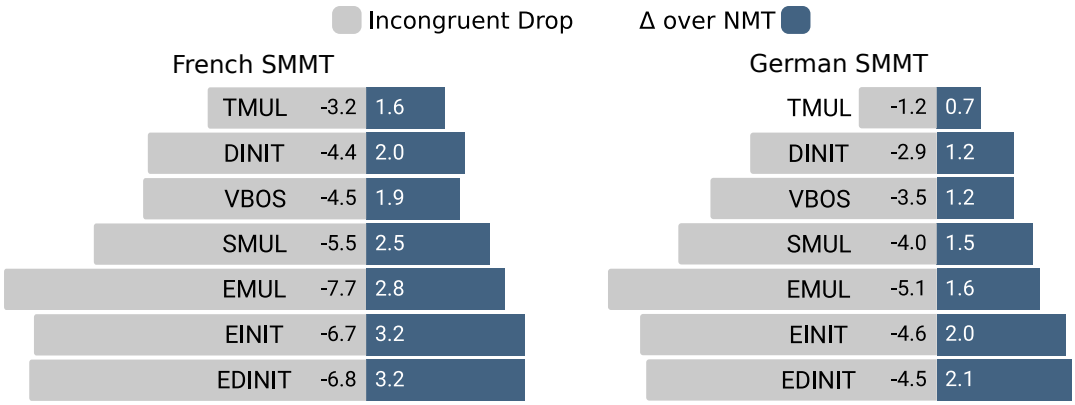


Figure 8.3: Entity masking results for German and French SMMTs: The boundary between the colored bars represents the METEOR score of the given MMT system.

dress" in Table 8.3. In these cases, we only replace the head noun (dress in this case) and leave the other words (blue) intact. The entity masking method is an aggressive degradation as it masks 26.2% of the words in both the `train` and the `test2016`. In terms of sentence statistics, this results in a training and test set where almost all the sentences contain at least one OOV token, with the average OOV per sentence being 3.4. Only 11 training sentences are not affected by this process. Unlike the progressive variant, entity masking guarantees systematic removal of visual information from source sentences since the originally annotated entities are concrete nouns.

Results

We conduct an extensive set of experiments and train all SMMT and AMMT variants for both English→German and English→French tasks. We then compute the congruent and incongruent METEOR scores across the three runs of each model. Finally, we compute the relative gains of each MMT over the **masked** NMT (Δ over NMT).

Figure 8.3 visualizes the results of German and French **SMMT** systems. We first notice that the encoder side interactions benefit the most from the visual modality unlike the target side interaction methods `TMUL`, `VBOS` and `DINIT` which seem quite ineffective. We observe critical performance drops with incongruent decoding, suggesting that the visual modality is now much more important than previously demonstrated (Elliott, 2018). In fact, a large multimodal gain is always coupled with a large incongruent drop.

As for the **AMMT** experiments, we present the incongruent drops and the multimodal gains for the concatenative variants as they tend to obtain slightly better METEOR scores on average when compared to their additive versions (Figure 8.4). The results suggest that the uniform visual attention (UVA) along with the shared attention system `SS-CAT`, benefit less from the visual modality.

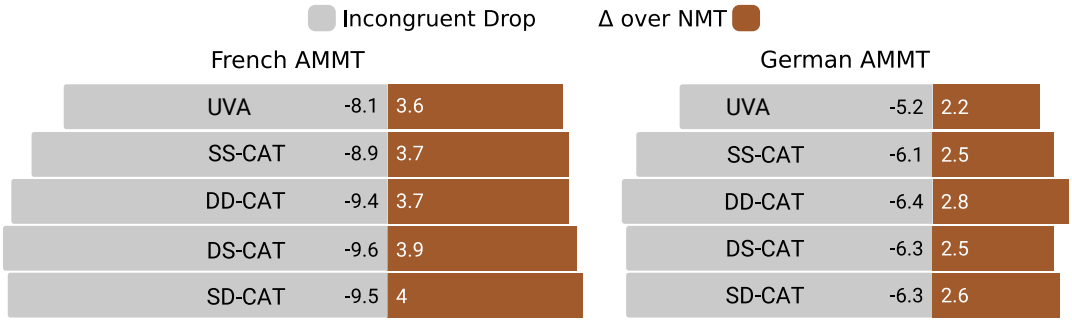


Figure 8.4: Entity masking results for German and French AMMTs: The boundary between the colored bars represents the METEOR score of the given MMT system.

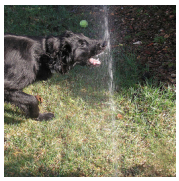
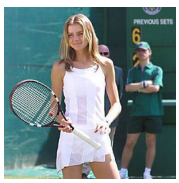
	SRC: a [v] drinks [v] outside on the [v] REF: un **chien** boit de **l'eau** dehors sur **l'herbe** *(a dog drinks water outside on the grass)* NMT: un homme boit du vin dehors sur le trottoir *(a man drinks wine outside on the sidewalk)* AMMT: un **chien** boit de **l'eau** dehors sur **l'herbe** INC: un homme boit des fleurs dehors sur l'herbe *(a man drinks flowers outside on the grass)*
	SRC: a [v] turns on the [v] to pursue a flying [v] REF: un **chien** tourne sur **l'herbe** pour poursuivre une balle en l'air *(a dog turns on the grass to chase a ball in the air)* NMT: un homme tourne sur la plage pour attraper un frisbee volant *(a man turns on the beach to catch a flying frisbee)* AMMT: un **chien** tourne sur **l'herbe** pour attraper un frisbee volant *(a dog turns on the grass to catch a flying frisbee)* INC: une femme se retourne sur le trottoir pour faire un objet volant *(a woman turns around on the sidewalk to make a flying object)*
	SRC: a young [v] in [v] holding a tennis [v] REF: **une** jeune **femme** en **blanc** tenant une raquette de tennis *(a young girl in white holding a tennis racket)* NMT: un jeune garçon en bleu tenant une raquette de tennis *(a young boy in blue holding a tennis racket)* AMMT: **une** jeune **femme** en **blanc** tenant une raquette de tennis INC: un jeune homme en bleu tenant une balle de tennis *(a young man in blue holding a tennis ball)*

Table 8.5: Entity masking examples from English→French models: underlined and bold words highlight bad and **good** lexical choices, respectively. English translations are provided in parentheses. The red INC lines are incongruent AMMT outputs.

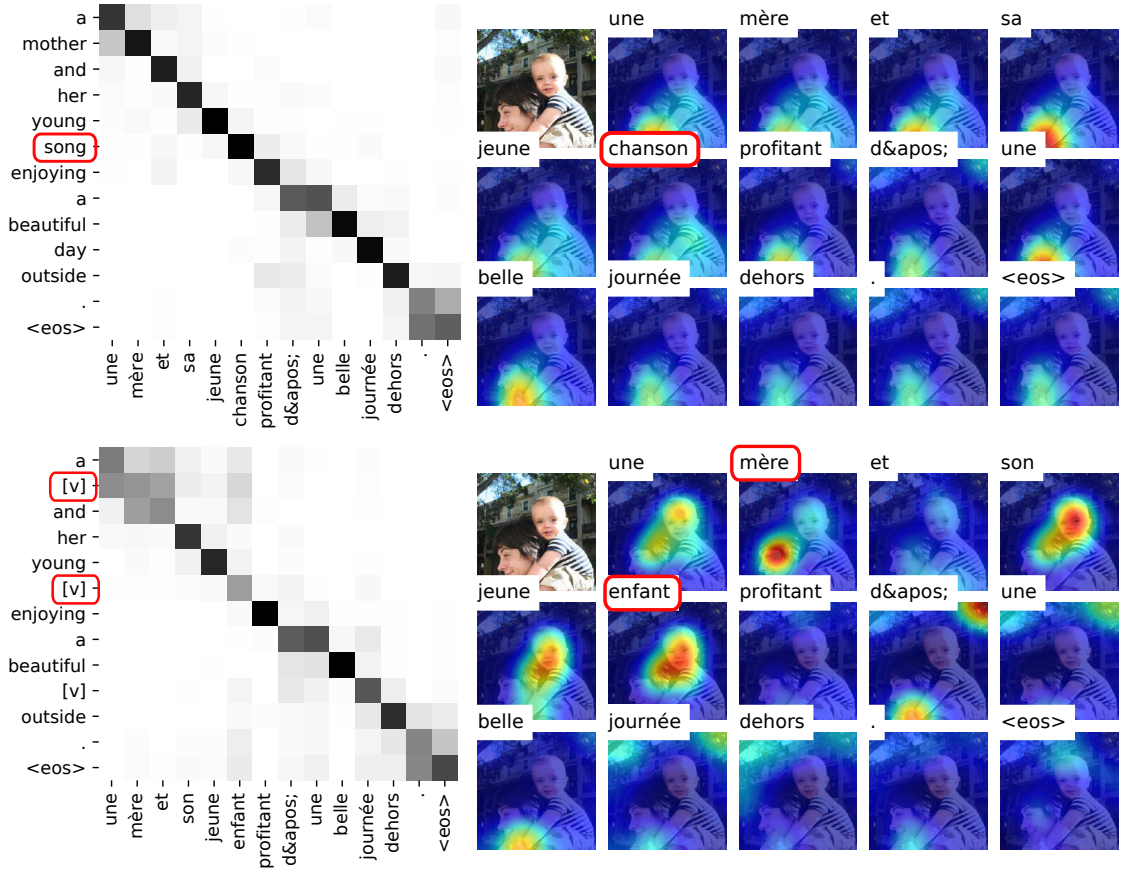


Figure 8.5: The impact of source degradation to visual attention: (top) Non-masked MMT translates the misspelled "son" (song → chanson) while the masked MMT (bottom) performs a correct translation ([v]→ enfant) by exploiting the visual modality.

For English→French, DS-CAT and SD-CAT systems perform the best on average while for German the completely dedicated attention DD-CAT obtains the best improvement. However, it should be noted that the differences among the AMMT variants are small: The best French system is significantly better than two AMMTs (p-value ≤ 0.05) while for German, the best system is not significantly better than the other variants. When compared to SMMT systems, the best AMMT for German and French significantly improves over the EDINIT systems with 0.7 and 0.8 METEOR, respectively. In overall, we can say that an AMMT with at least some level of modality specific dedication, outperforms any other MMT variants including the SMMT ones.

Finally, Table 8.5 provides qualitative examples (more examples are provided in Appendix A) for entity masking experiments where the selected AMMT is the DD-CAT system. Similar to progressive masking, we observe that the MMT system successfully fills in the blanks with the help of the visual modality. When incongruently decoded, the AMMT model mostly behaves like the masked NMT and loses its ability to produce visually coherent sentences.

A manual inspection of visual attention maps produced by the default, non-masked MMTs (chapter 7) and entity masked MMTs reveals that the attention is much more plausible and "active" in the latter. An interesting example is given in Figure 8.5 where the masked MMT attends to the correct region of the image and successfully translates a masked word that was otherwise a spelling mistake in the source sentence ("son" written as "song"). However, the non-masked MMT attention is stuck at the lower right portion of the image. All masked MMT models are able to correctly translate this sentence unlike the non-masked ones that blindly rely on the spelling mistake.

8.3 Summary

In this chapter, I presented an in-depth study on the potential contribution of images for MMT. Specifically, I analysed the behavior of our SMMT and AMMT systems under two degradation schemes where information is systematically removed from source sentences. The results show that the proposed SMMT and AMMT models successfully exploit the visual modality when the linguistic context is scarce, but tend to be less sensitive to the images when exposed to complete sentences during training. In the latter case, the language signal turns out to be sufficient to accomplish the task and dominates the visual modality. We think that this dominance is expected since NMT is quite good at performing sequence-to-sequence transduction, leaving no space to the externally injected visual signal. Interestingly, this behavior corroborates the seminal work of Colavita (1974) in Psychophysics where it has been demonstrated that visual stimuli dominate over the auditory stimuli when humans are asked to perform a simple audiovisual discrimination task. In the light of these, it is likely that the majority of the current state-of-the-art models are affected by this dominance since the adversarial evaluation did not reveal any signs of complete collapse in the literature (Elliott, 2018; Grönroos et al., 2018) and also for our SMMT and AMMT models. We thus suspect that – at least for the Multi30K dataset – the consequences of integrating the visual modality are secondary, reminding previous work about the influence of random perturbations to DNN training: Gulcehre et al. (2016) deliberately inject random noise into non-linear activation functions which in turn improves the dynamics of the gradients while Neelakantan et al. (2015) shows evidence of improved generalization when adding gaussian noise to the gradients.

Finally, the degradation experiments also reveal that the attentive models which integrate spatial features, perform significantly better than the simple models that use global visual features. Our investigation also suggests that visual grounding can increase the robustness of MT systems by mitigating input noise such as errors in the source text.

CHAPTER 9

Conclusion & Discussion

In this study, we concentrated on designing novel NMT systems that can leverage contextual information from auxiliary modalities. For this purpose, we specifically worked with the Multi30K dataset (Elliott et al., 2016) which contains images and their translated descriptions. The visual context can be beneficial to this dataset as it can encourage MT systems to apply visually guided word sense disambiguation, missing word imputation, or gender marking between gender-neutral and gendered languages. Besides being an interesting task on its own, a successful MMT system is also important to foster research on multimodal language understanding in general.

We mainly explored two different multimodal approaches, which further determine the type of visual features used to represent the images. First, we extracted global visual features from state-of-the-art pre-trained CNN models and experimented with grounding the intermediate components of an NMT with vectorial image representations. For the second type of models, we again took advantage of pre-trained CNNs, but this time we extracted convolutional features that preserve spatial information unlike the previous global representations. These richer features are then integrated into a novel "multimodal attention" mechanism in the NMT, with the purpose of guiding the decoder to look at the image when translating a given sentence.

Upon extensive analyses based on automatic evaluation metrics, we observed moderate to significant improvements for English→German but the same did not hold for English→French. For the multimodal attention based models, we quantitatively and qualitatively showed that L_2 normalization of the features is crucial for the visual attention to be effective. To better understand to what extent the visual modality is taken into account by the models, we conducted the adversarial evaluation protocol (Elliott, 2018) and noticed that most of the models barely respond to incongruent decoding, some of them even mildly improving similar to what has been observed by Grönroos et al.

(2018). This brought up the question of whether the proposed MMT systems are even architecturally capable of leveraging the visual modality or not. To that end, we artificially created scenarios where the visual modality is "required" to perform well on the task, such as systematically removing suffixes or masking out visually depictable nouns from the source sentences. This final set of experiments clearly showed that the images are indeed taken into account by both global feature and spatial feature based MMT with the latter performing significantly better than the former. We have also found evidence that the visual grounding can improve the robustness of MT systems by mitigating input noise such as spelling errors.

We now briefly discuss several perspectives and insights about the next steps in multimodal language learning.

Better MMT Approaches

Borrowing from the insights of this work, I think that it would be interesting to design MMT systems which integrate a sort of message passing mechanism across modalities: The modality attentions can then be guided by entropy-based gating mechanisms for example, so that at each timestep the more confident modality can take over the other one when computing the attended context.

I also believe that there remains a lot to explore in terms of handling OOV words at inference time. Although subword segmentation mitigates the problem in theory, the issue is still there: Consider the source sentence "a path leads to a pagoda" where the OOV token "pagoda" gets segmented into "p@@ ag@@ o@@ da" using the BPE algorithm. Although the token is no longer an OOV, it is practically impossible for the current NMT and MMT models to generate a sequence of subwords that would form the French translation "pagode". In fact, BPE even encourages hallucination here as the model would be forced to translate the sequence of source embeddings [p@@, ag@@, o@@, da] into something[1]. On the other hand, if we were to keep the token as an OOV, the model could detect it and attempt to refer to some kind of multimodal knowledge base in order to fetch most probable candidate words that would be integrated into the decoding logic. It may be possible to construct this knowledge base using state-of-the-art pre-trained word embeddings and visual features in ways similar to visual bilingual lexicon induction methods (Kiela et al., 2015).

[1] We discovered French hallucinations such as "pylessive" and "limetière" when NMT and MMT systems translate the word "pagoda".

Better MMT Evaluation

During the extensive quantitative and qualitative analyses, we were often faced with the question of how should the effectiveness of an MMT be evaluated. Our sentence level analyses in the previous chapters clearly showed one thing: Even retraining a model with different random initialization yields substantially different translations for 70% to 85% of the test set. These abrupt shifts from one run to another can be a confounding factor that may hide small but valuable improvements due to the multimodality. This is also not completely mitigated by human evaluation as humans will also show different levels of appreciations for this intrinsic translation variance: For example, we previously saw how the incongruent decoding replaced "footballspieler" with the wrong "football-spieler" version, an effect unlikely to be related to multimodality. In light of these, I believe that the MMT systems should be evaluated with custom fine-grained protocols instead of corpus level metrics. An example of this was proposed for MT evaluation through a "challenge set" which probes the abilities of state-of-the-art MT systems in terms of several global and language-specific linguistic phenomena (Isabelle et al., 2017).

New Datasets & Tasks

Although a more challenging test set with ambiguous verb uses was published (Elliott et al., 2017) for Multi30K, there has not been any exciting results showing substantial improvements over NMT baselines. I believe that this makes sense as we do not know to what extent the training set of Multi30K is affected by contextual ambiguities *i.e.* if the models are never challenged with multiple senses of a verb during training, it is quite unrealistic that they will be grounded to resolve such ambiguities at test time. The fact that the state-of-the-art baselines converged to extremely high BLEU and METEOR scores also suggest that we may need more challenging datasets for which the auxiliary modalities are vital. To that end, we proposed a new multimodal dataset called *How2* which consists of more than 70K instructional videos with English subtitles and their crowd-sourced Portuguese translations (Sanabria et al., 2018). The unique combination of video, speech and bilingual subtitles allow the exploration of many tasks such as automatic speech recognition (ASR), speech translation and machine translation. For each one of them, a multimodal variant exists where the auxiliary modality can be visual and/or auditory.

Multimodal Speech Recognition

In the context of automatic speech recognition (ASR), the presence of a synchronized video stream of the narrator enables *lipreading* (Chung et al., 2017), a technique to reduce the effect of ambient noise. This approach can be defined as a *local grounding* since the grounding happens between *phonemes* and their visual counterparts *visemes*. On the other hand, *global grounding* can always happen when the video consistently provides object, action and scene level cues correlated with the speech content as may be the case with the instructional videos of *How2* dataset. Here, visual cues from the recording environment (indoor vs outdoor) or the interaction between salient objects (people, instruments, vehicles, tools and equipments) can be exploited by the recognizer in various ways to learn a better acoustic and/or language model. In Caglayan et al. (2019b), we experimented with our `EINIT`, `DINIT`, `EDINIT` and `VBOS` grounding methods (Chapter 6), with the global visual features being extracted using the middle frame of video segments. We obtained moderate improvements of up to 1% reduction in word error rate using the `EDINIT` approach with other approaches performing mildly worse than it, similar to our MMT results in this work.

Simultaneous Contextual MT

In chapter 8, we showed the effectiveness of the visual modality when sentence suffixes are systematically removed from the language input. This is an interesting insight which encourages us towards extending the currently available simultaneous NMT systems (Cho and Esipova, 2016; Gu et al., 2017; Dalvi et al., 2018) with the visual modality. We believe this is a nice way of leveraging the multimodality which would potentially decrease the source context delay in simultaneous MT.

Selected Publications

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016a. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 627–633.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016b. Multimodal attention for neural machine translation. *Computing Research Repository* arXiv:1609.03976.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 432–439.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. NMTPY: A flexible toolkit for advanced neural machine translation systems. *Prague Bull. Math. Linguistics* 109:15–28.

Ozan Caglayan, Adrien Bardet, Fethi Bougares, Loïc Barrault, Kai Wang, Marc Masana, Luis Herranz, and Joost van de Weijer. 2018. LIUM-CVC submissions for WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 603–608.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (NeurIPS 2018)*.

Ozan Caglayan, Ramon Sanabria, Shruti Palaskar, Loïc Barrault, and Florian Metze. 2019b. Multimodal grounding for sequence-to-sequence speech recognition. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019a. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4159–4170.

APPENDIX A

Additional Masking Examples

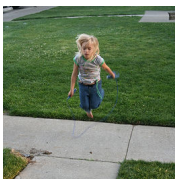
	SRC: a [v] in a red [v] plays in the [v] NMT: un garçon en t-shirt rouge joue dans la neige *(a boy in a red t-shirt plays in the snow)* AMMT: un garçon en **maillot de bain** rouge joue dans **l'eau** REF: un garçon en **maillot de bain** rouge joue dans **l'eau** *(a boy in a red swimsuit plays in the water)*
	SRC: two [v] are driving on a [v] NMT: deux hommes font du vélo sur une route *(two men riding bicycles on a road)* AMMT: deux **voitures roulent sur une piste** *(two cars driving on a track/circuit)* REF: deux **voitures roulent sur** un circuit
	SRC: a [v] jumping [v] on a [v] near a parking [v] NMT: un homme sautant à cheval sur une plage près d'un parking *(a man jumping on a beach near a parking lot)* AMMT: une **fille** sautant à la **corde** sur un **trottoir** près d'un parking REF: une **fille** sautant à la **corde** sur un **trottoir** près d'un parking *(a girl jumping rope on a sidewalk near a parking lot)*

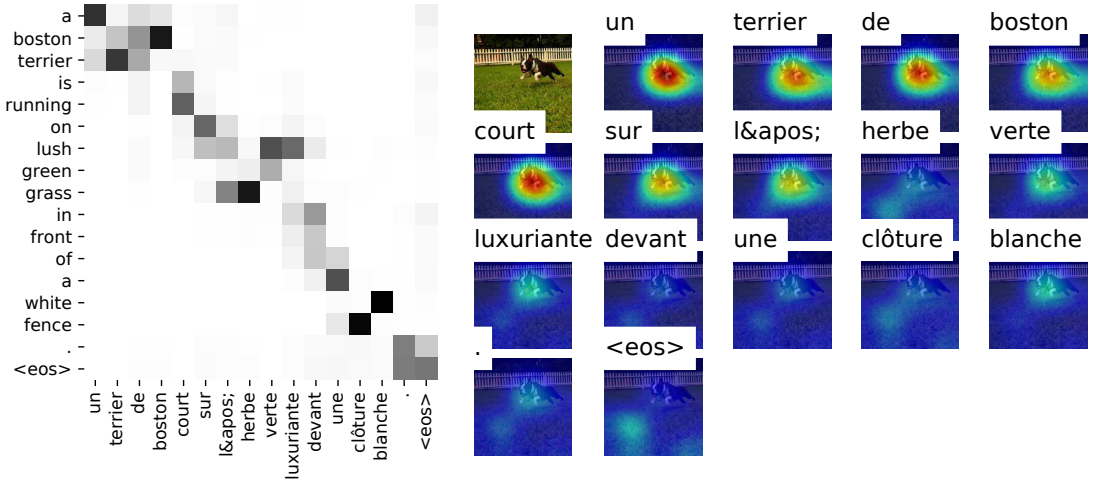
Table A.1: Additional entity masking examples: underlined and bold words highlight bad and **good** lexical choices, respectively. English translations are provided in parentheses.

	SRC: a jockey riding his [v][v] NMT: un jockey sur son vélo *(a jockey on his bike)* AMMT: un jockey sur son **cheval** REF: un jockey sur son **cheval** *(a jockey on his horse)*
	SRC: a fishing net on the deck of a [v][v] NMT: un filet de pêche sur la terrasse d'un bâtiment *(a fishing net on the terrace of a building)* AMMT: un filet de pêche sur le **pont d'un bateau** *(a fishing net on the deck of a boat)* REF: un filet de pêche sur le **pont d'un bateau** rouge *(a fishing net on the deck of a red boat)*
	SRC: girls are playing a [v][v][v] NMT: des filles jouent à un jeu de cartes *(girls are playing a card game)* AMMT: des filles jouent un **match de football** REF: des filles jouent un **match de football** *(girls are playing a football match)*
	SRC: a child [v][v][v][v][v][v] NMT: un enfant avec des lunettes de soleil en train de jouer au tennis *(a child with sunglasses playing tennis)* AMMT: un enfant **est debout dans un champ de fleurs** *(a child is standing in field of flowers)* REF: un enfant **dans un champ de tulipes** *(a child in a field of tulips)*

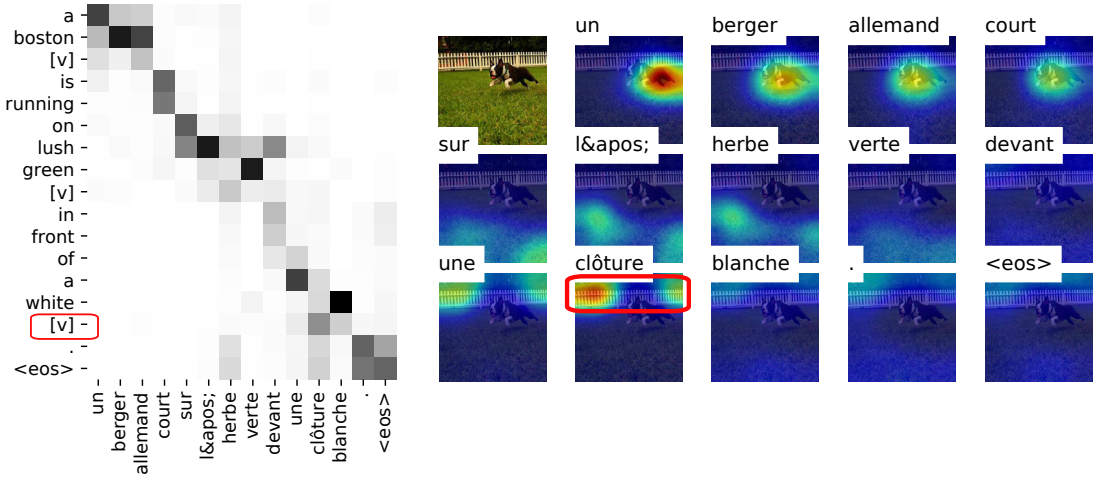
Table A.2: Additional progressive masking examples: underlined and bold words highlight bad and **good** lexical choices, respectively. English translations are provided in parentheses.

$\mathcal{D}_0$	SRC:	[v]...
	NMT:	un homme vêtu d'un t-shirt bleu et d'un jean est assis sur un banc
		a man wearing a blue t-shirt and a jean is sitting on a bank
	AMMT:	une femme en maillot de bain rouge est assise sur un rocher au bord de l'eau
		a woman in a red swimsuit is sitting on a rock at the seaside
$\mathcal{D}_2$	SRC:	an older [v]...
	NMT:	un vieil homme vêtu d'un t-shirt blanc et d'un jean est assis sur un banc
		an older man wearing a blue t-shirt and a jean is sitting on a bank
	AMMT:	une femme âgée vêtue d'un maillot de bain rouge est assis sur un rocher au bord de l'eau
		an older woman wearing a red swimsuit is sitting on a rock at the seaside
$\mathcal{D}_4$	SRC:	an older woman in [v]...
	NMT:	une femme âgée en t-shirt bleu est assise sur un banc dans un parc
		an older woman in a blue t-shirt is sitting on a bank in the park
	AMMT:	une femme âgée en maillot de bain rose est assise sur un rocher au bord de l'eau
		an older woman in a pink swimsuit is sitting on a rock at the seaside
$\mathcal{D}_6$	SRC:	an older woman in a bikini [v]...
	NMT:	une femme âgée en bikini est assise sur un banc dans un parc
		an older woman in bikini is sitting on a bank in the park
	AMMT:	une femme âgée en bikini est assise sur un rocher au bord de l'eau
		an older woman in a swimsuit is sitting on a rock at the seaside
$\mathcal{D}_8$	SRC:	an older woman in a bikini is sunbathing [v]...
	NMT:	une femme âgée en bikini fait un bain de soleil sur un trottoir en ville
		an older woman in bikini is sunbathing on a sidewalk in the city
	AMMT:	une femme âgée en bikini fait un salto arrière sur la plage
		an older woman in bikini performs a back loop in the beach
$\mathcal{D}_{10}$	SRC:	an older woman in a bikini is sunbathing on a [v]...
	NMT:	une femme âgée en bikini est en train de nager sur un banc dans un parc
		an older woman in bikini is swimming on a bank in the park
	AMMT:	une femme âgée en bikini fait du soleil sur un rocher au bord de l'eau
		an older woman in bikini is sunbathing on a rock at the seaside
$\mathcal{D}_{12}$	SRC:	an older woman in a bikini is sunbathing on a rock by [v]...
	NMT:	une femme âgée en bikini nage sur un rocher au bord de l'eau
		an older woman in bikini swims on a rock at the seaside
	AMMT:	une femme âgée en bikini fait du soleil sur un rocher au bord de l'eau
		an older woman in bikini is sunbathing on a rock at the seaside
$\mathcal{D}$	SRC:	an older woman in a bikini is sunbathing on a rock by the ocean
	NMT:	une femme âgée en bikini fait du soleil sur un rocher au bord de l'océan
	AMMT:	une femme âgée en bikini fait du soleil sur un rocher au bord de l'océan
		an older woman in bikini is sunbathing on a rock at the seaside

Table A.3: Successive outputs from progressively masked NMT and AMMT.



(a) Non-masked MMT



(b) Entity-masked MMT

Figure A.1: Additional visual attention example for entity masking where *terrier*, *grass* and *fence* are dropped from the source sentence: (a) Non-masked MMT is not able to shift attention from the salient *dog* to the *grass* and *fence*, (b) the attention produced by the masked MMT first shifts to the background area while translating "on lush green [v]" then focuses on the *fence*.

Bibliography

Rana Abu-Zhaya, Amanda Seidl, Ruth Tincoff, and Alejandrina Cristia. 2017. Building a multimodal lexicon: Lessons from infants' learning of body part words. In *Proc. GLU 2017 International Workshop on Grounding Language Understanding*. pages 18–21.

Walid Aransa, Holger Schwenk, and Loïc Barrault. 2015. Improving continuous space language models using auxiliary features. In *Proceedings of the 12th International Workshop on Spoken Language Translation*. Da Nang, Vietnam, pages 151–158.

Devansh Arpit and Yoshua Bengio. 2019. The benefits of over-parameterization at initialization in deep relu networks. *Computing Research Repository* arXiv:1901.03611.

Hasan Sait Arslan, Mark Fishel, and Gholamreza Anbarjafari. 2018. Doubly attentive transformer machine translation. *Computing Research Repository* arXiv:1807.11605.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computing Research Repository* arXiv:1409.0473. Version 7.

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2017. Multimodal machine learning: A survey and taxonomy. *CoRR* abs/1705.09406.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Belgium, Brussels, pages 308–327.

Rachel Bawden. 2018. *Going beyond the sentence : Contextual Machine Translation of Dialogue*. Theses, Université Paris-Saclay.

Atilim Güneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2017. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* 18(1):5595–5637.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 1171–1179.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Yoshua Bengio, Patrick Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5(2):157–166.

Shane Bergsma and Benjamin Van Durme. 2011. Learning bilingual lexicons using the visual similarity of labeled web images. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*. AAAI Press, IJCAI'11, pages 1764–1769.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5(1):135–146.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Belgium, Brussels, pages 272–307.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc., USA, NIPS'16, pages 4356–4364.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013. Audio chord recognition with recurrent neural networks. In *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*. pages 335–340.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1442–1451.

Franck Burlot, Mercedes García-Martínez, Loïc Barrault, Fethi Bougares, and François Yvon. 2017. Word representations in factored neural machine translation. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics, Copenhagen, Denmark, pages 20–31.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 432–439.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016a. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 627–633.

Ozan Caglayan, Adrien Bardet, Fethi Bougares, Loïc Barrault, Kai Wang, Marc Masana, Luis Herranz, and Joost van de Weijer. 2018. LIUM-CVC submissions for WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 603–608.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016b. Multimodal attention for neural machine translation. *Computing Research Repository* arXiv:1609.03976.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. NMTPY: A flexible toolkit for advanced neural machine translation systems. *Prague Bull. Math. Linguistics* 109:15–28.

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019a. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4159–4170.

Ozan Caglayan, Ramon Sanabria, Shruti Palaskar, Loïc Barrault, and Florian Metze. 2019b. Multimodal grounding for sequence-to-sequence speech recognition. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

Iacer Calixto, Koel Dutta Chowdhury, and Qun Liu. 2017a. DCU system report on the WMT 2017 multi-modal machine translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 440–444.

Iacer Calixto, Desmond Elliott, and Stella Frank. 2016. DCU-UvA multimodal MT system report. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 634–638.

Iacer Calixto and Qun Liu. 2017. Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 992–1003.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017b. 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)*. Association for Computational Linguistics, pages 1913–1924.

Rich Caruana. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

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

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *Computing Research Repository* arXiv:1606.02012.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. 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*. Association for Computational Linguistics, Doha, Qatar, pages 103–111.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. 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)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734.

Grzegorz Chrupała, Ákos Kádár, and Afra Alishahi. 2015. 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). Association for Computational Linguistics, Beijing, China, pages 112–118.

Joon Son Chung, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. pages 3444–3453.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Computing Research Repository* arXiv:1412.3555.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 176–181.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (ELUs). *Computing Research Repository* arXiv:1511.07289.

Francis B. Colavita. 1974. Human sensory dominance. *Perception & Psychophysics* 16 (2):409–412.

George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2(4):303–314.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation 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 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 493–499.

Emile Delavenay and Katharine M Delavenay. 1960. *An introduction to machine translation*. Thames and Hudson London.

Jean-Benoit Delbrouck and Stéphane Dupont. 2017a. Modulating and attending the source image during encoding improves multimodal translation. *Computing Research Repository* arXiv:1712.03449.

Jean-Benoit Delbrouck and Stéphane Dupont. 2017b. Multimodal compact bilinear pooling for multimodal neural machine translation. *Computing Research Repository* arXiv:1703.08084.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pages 248–255.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pages 376–380.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1370–1380.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1723–1732.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

John Duselis, Michael Hutt, Jeremy Gwinnup, James Davis, and Joshua Sandvick. 2017. The AFRL-OSU WMT17 multimodal translation system: An image processing approach. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 445–449.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 2974–2978.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 215–233.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision*

and Language. Association for Computational Linguistics, Berlin, Germany, pages 70–74.

Desmond Elliott and Àkos Kádár. 2017. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Asian Federation of Natural Language Processing, pages 130–141.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Marc O. Ernst and Martin S. Banks. 2002. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* 415:429.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Comput. Speech Lang.* 45(C):236–252.

John Rupert Firth. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, Philological Society, Oxford. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Stella Frank, Desmond Elliott, and Lucia Specia. 2018. Assessing multilingual multimodal image description: Studies of native speaker preferences and translator choices. *Natural Language Engineering* 24(3):393–413.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 457–468.

Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36(4):193–202.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc., USA, NIPS'16, pages 1027–1035.

Mercedes García-Martínez, Ozan Caglayan, Walid Aransa, Adrien Bardet, Fethi Bougares, and Loïc Barrault. 2017. Lium machine translation systems for WMT17

news translation task. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics, Copenhagen, Denmark, pages 288–295.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. JMLR.org, ICML'17, pages 1243–1252.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. PMLR, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering* 23(1):3–30.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *Computing Research Repository* arXiv:1308.0850.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2015. Lstm: A search space odyssey. *Computing Research Repository* arXiv:1503.04069.

Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, and Raúl Vázquez. 2018. The MeMAD submission to the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 609–617.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 1053–1062.

Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings*

of The 33rd International Conference on Machine Learning. PMLR, New York, New York, USA, volume 48 of *Proceedings of Machine Learning Research*, pages 3059–3068.

Jeremy Gwinnup, Joshua Sandvick, Michael Hutt, Grant Erdmann, John Duselis, and James Davis. 2018. The AFRL-Ohio State WMT18 multimodal system: Combining visual with traditional. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 618–621.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation*. Seattle, USA.

Zellig S. Harris. 1954. Distributional structure. *¡i¿WORD¡/i¿* 10(2-3):146–162.

Kaiming He, Zhang Xiangyu, Ren Shaoqing, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*. pages 1026–1034.

Kaiming He, Zhang Xiangyu, Ren Shaoqing, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 770–778.

Jindřich Helcl and Jindřich Libovický. 2017. CUNI system for the WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 450–457.

Jindřich Helcl, Jindřich Libovický, and Dusan Varis. 2018. CUNI system for the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 622–629.

Sepp Hochreiter. 1998. Recurrent neural net learning and vanishing gradient. *International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems* 6(2):107–116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9(8):1735–1780.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. 2017a. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations (ICLR)*.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017b. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. 2016. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 639–645.

David H Hubel and Torsten N Wiesel. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology* 160(1):106–154.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *Computing Research Repository* arXiv:1611.01462.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*. pages 448–456.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2486–2496.

Jana M Iverson and Susan Goldin-Meadow. 2005. Gesture paves the way for language development. *Psychological science* 16(5):367–371.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Computing Research Repository* arXiv:1611.04558.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, ICML'15, pages 2342–2350.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models.

In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1700–1709.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering. *Computing Research Repository* arXiv:1704.03162.

Douwe Kiela, Ivan Vulić, and Stephen Clark. 2015. Visual bilingual lexicon induction with transferred convnet features. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 148–158.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository* arXiv:1412.6980.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *Computing Research Repository* arXiv:1411.2539.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 971–980.

R. Kneser and H. Ney. 1995. Improved backing-off for M-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*. volume 1, pages 181–184 vol.1.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Meeting of the Association for Computational Linguistics*. pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation.* Association for Computational Linguistics, Vancouver, pages 28–39.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '03, pages 48–54.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., pages 1097–1105.

Anders Krogh and John A. Hertz. 1992. A simple weight decay can improve generalization. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, Morgan-Kaufmann, pages 950–957.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, pages 66–71.

Chiraag Lala, Pranava Swaroop Madhyastha, Carolina Scarton, and Lucia Specia. 2018. Sheffield submissions for WMT18 multimodal translation shared task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 630–637.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Stroudsburg, PA, USA, StatMT '07, pages 228–231.

Yann LeCun. 1988. A theoretical framework for back-propagation. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann, CMU, Pittsburgh, Pa, pages 21–28.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Jindřich Libovický and Jindřich Helcl. 2017. 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)*. Association for Computational Linguistics, pages 196–202.

Jindrich Libovický, Shruti Palaskar, Spandana Gella, and Florian Metze. 2018. Multimodal abstractive summarization of opendomain videos. In *NeurIPS Workshop on Visually Grounded Interaction and Language (ViGIL)*.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI system for WMT16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 646–654.

Jindřich Libovický, Jindřich Helcl, and David Mareček. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, Belgium, Brussels, pages 253–260.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *Computing Research Repository* arXiv:1511.06114.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015b. Effective approaches to attention-based neural machine translation. *Computing Research Repository* arXiv:1508.04025.

Mingbo Ma, Dapeng Li, Kai Zhao, and Liang Huang. 2017. OSU multimodal machine translation system report. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 465–469.

Pranava Swaroop Madhyastha, Josiah Wang, and Lucia Specia. 2017. Sheffield MultiMT: Using object posterior predictions for multimodal machine translation. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 470–476.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (m-rnn). In *International Conference on Learning Representations (ICLR)*.

James Martens. 2010. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, USA, ICML'10, pages 735–742.

Warren S. McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5(4):115–133.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computing Research Repository* arXiv:1301.3781.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*. volume 2, page 3.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. 2015. Adding gradient noise improves learning for very deep networks. *Computing Research Repository* arXiv:1511.06807.

Eric H. Nyberg and Teruko Mitamura. 1992. The kant system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 3*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '92, pages 1069–1073.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '03, pages 160–167.

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

Razvan Pascanu, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. In *International Conference on Learning Representations (ICLR)*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*. PMLR, Atlanta, Georgia, USA, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques, Long Beach, CA, US*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)*. pages 2641–2649.

Marcelo O. R. Prates, Pedro H. Avelar, and Luís C. Lamb. 2019. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications* .

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *Computing Research Repository* arXiv:1608.05859.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. 2014. CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 512–519.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *International Conference on Learning Representations*.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Paper*. Association for Computational Linguistics, Copenhagen, Denmark, pages 11–19.

Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* pages 65–386.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323:533–.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (NeurIPS 2018)*.

Ramon Sanabria, Shruti Palaskar, and Florian Metze. 2019. CMU Sinbad's submission for the DSTC7 AVSD challenge. In *DSTC7 at AAAI2019 workshop*.

Andrew M Saxe, James L McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *Computing Research Repository* arXiv:1312.6120.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61:85 – 117.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Holger Schwenk. 2010. Continuous space language models for statistical machine translation. In *The Prague Bulletin of Mathematical Linguistics, (93):137–146.*.

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012: Posters*. The COLING 2012 Organizing Committee, Mumbai, India, pages 1071–1080.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2016. Recurrent dropout without memory loss. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, pages 1757–1766.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch-Mayne, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. *Nematus: a Toolkit for Neural Machine Translation*, Association for Computational Linguistics (ACL), pages 65–68.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725.

Kashif Shah, Josiah Wang, and Lucia Specia. 2016. SHEF-Multimodal: Grounding machine translation on images. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 660–665.

Carina Silberer and Mirella Lapata. 2012. Grounded models of semantic representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, pages 1423–1433.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository* arXiv:1409.1556.

Xingyi Song, Trevor Cohn, and Lucia Specia. 2013. BLEU deconstructed: Designing a better MT evaluation metric. *International Journal of Computational Linguistics and Applications* 4(2):29.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 543–553.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Barry E. Stein, Terrence R. Stanford, and Benjamin A. Rowland. 2009. The neural basis of multisensory integration in the midbrain: Its organization and maturation. *Hearing Research* 258(1):4 – 15. Multisensory integration in auditory and auditory-related areas of cortex.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *Computing Research Repository* arXiv:1605.02688.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*. Association for Computational Linguistics, Copenhagen, Denmark, pages 82–92.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pages 3156–3164.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1264–1274.

Kevin Vythelingum, Yannick Estève, and Olivier Rosec. 2018. Acoustic-dependent phonemic transcription for text-to-speech synthesis. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*. pages 2489–2493.

Paul J. Werbos. 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. thesis, Harvard University.

Paul J. Werbos. 1982. Applications of advances in nonlinear sensitivity analysis. In R. F. Drenick and F. Kozin, editors, *System Modeling and Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 762–770.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015a. Empirical evaluation of rectified activations in convolutional network. *Computing Research Repository* arXiv:1505.00853.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015b. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings, pages 2048–2057.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2:67–78.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *Computing Research Repository* arXiv:1212.5701.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*. Springer International Publishing, Cham, pages 818–833.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2017. NICT-NAIST system for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Copenhagen, Denmark, pages 477–482.

Renjie Zheng, Yilin Yang, Mingbo Ma, and Liang Huang. 2018. Ensemble sequence level training for multimodal MT: OSU-Baidu WMT18 multimodal machine translation system report. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels, pages 638–642.

Mingyang Zhou, Runxiang Cheng, Yong Jae Lee, and Zhou Yu. 2018. A visual attention grounding neural model for multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 3643–3653.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 30–34.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Le Mans Université

Titre : Traduction Automatique Multimodale

Mots clés : Multimodalité, Traduction Automatique, Réseau de Neurones Profonds, Traitement Automatique des Langues

Resumé : La traduction automatique vise à traduire des documents d'une langue à une autre sans l'intervention humaine. Avec l'apparition des réseaux de neurones profonds (DNN), la traduction automatique neuronale (NMT) a commencé à dominer le domaine, atteignant l'état de l'art pour de nombreuses langues. NMT a également ravivé l'intérêt pour la traduction basée sur l'*interlangue* grâce à la manière dont elle place la tâche dans un cadre encodeur-décodeur en passant par des représentations latentes. Combiné avec la flexibilité architecturale des DNN, ce cadre a aussi ouvert une piste de recherche sur la multimodalité, ayant pour but d'enrichir les représentations latentes avec d'autres modalités telles que la vision ou la parole, par exemple. Cette thèse se concentre sur la traduction automatique multimodale (MMT) en intégrant la vision comme une modalité secondaire afin d'obtenir une meilleure compréhension du langage, ancrée de façon visuelle. J'ai travaillé spécifiquement avec un ensemble de données contenant des images et leurs descriptions traduites, où le contexte visuel peut être utile pour désambiguïser le sens des mots polysémiques, imputer des mots manquants ou déterminer le genre lors de la traduction vers une langue ayant du genre grammatical comme avec l'anglais vers le français. Je propose deux approches principales pour intégrer la modalité visuelle : (i) un mécanisme d'attention multimodal qui apprend à prendre en compte les représentations latentes des phrases sources ainsi que les caractéristiques visuelles convolutives, (ii) une méthode qui utilise des caractéristiques visuelles globales pour amorcer les encodeurs et les décodeurs récurrents. Grâce à une évaluation automatique et humaine réalisée sur plusieurs paires de langues, les approches proposées se sont montrées bénéfiques. Enfin, je montre qu'en supprimant certaines informations linguistiques à travers la dégradation systématique des phrases sources, la véritable force des deux méthodes émerge en imputant avec succès les noms et les couleurs manquants. Elles peuvent même traduire lorsque des morceaux de phrases sources sont entièrement supprimés.

Title : Multimodal Machine Translation

Keywords : Multimodality, Machine Translation, Deep Neural Networks, Natural Language Processing

Abstract : Machine translation aims at automatically translating documents from one language to another without human intervention. With the advent of deep neural networks (DNN), neural approaches to machine translation started to dominate the field, reaching state-of-the-art performance in many languages. Neural machine translation (NMT) also revived the interest in *interlingual* machine translation due to how it naturally fits the task into an encoder-decoder framework which produces a translation by decoding a latent source representation. Combined with the architectural flexibility of DNNs, this framework paved the way for further research in multimodality with the objective of augmenting the latent representations with other modalities such as vision or speech, for example. This thesis focuses on a multimodal machine translation (MMT) framework that integrates a secondary visual modality to achieve better and visually grounded language understanding. I specifically worked with a dataset containing images and their translated descriptions, where visual context can be useful for word sense disambiguation, missing word imputation, or gender marking when translating from a language with gender-neutral nouns to one with grammatical gender system as is the case with English to French. I propose two main approaches to integrate the visual modality : (i) a multimodal attention mechanism that learns to take into account both sentence and convolutional visual representations, (ii) a method that uses global visual feature vectors to prime the sentence encoders and the decoders. Through automatic and human evaluation conducted on multiple language pairs, the proposed approaches were demonstrated to be beneficial. Finally, I further show that by systematically removing certain linguistic information from the input sentences, the true strength of both methods emerges as they successfully impute missing nouns, colors and can even translate when parts of the source sentences are completely removed.