Adversarial reconstruction for Multi-modal Machine Translation

Jean-Benoit Delbrouck and **Stéphane Dupont**
TCTS Lab, University of Mons, Belgium
{jean-benoit.delbrouck, stephane.dupont}@umons.ac.be

Abstract

Even with the growing interest in problems at the intersection of Computer Vision and Natural Language, grounding (i.e. identifying) the components of a structured description in an image still remains a challenging task. This contribution aims to propose a model which learns grounding by reconstructing the visual features for the Multi-modal translation task. Previous works have partially investigated standard approaches such as regression methods to approximate the reconstruction of a visual input. In this paper, we propose a different and novel approach which learns grounding by adversarial feedback. To do so, we modulate our network following the recent promising adversarial architectures and evaluate how the adversarial response from a visual reconstruction as an auxiliary task helps the model in its learning. We report the highest scores in term of BLEU and METEOR metrics on the different datasets.

1 Introduction

Problems combining vision and natural language processing are viewed as a difficult task. It requires to grasp and express low to high-level aspects of local and global areas in an image as well as their relationships. Visual attention-based neural decoder models (Xu et al., 2015; Karpathy and Li, 2015) have been widely adopted to solve such tasks. The attention focuses only on part of an image and integrates this spatial information into the multi-modal model pipeline. The model, that usually consists of a Recurrent Neural Network (RNN), encodes the linguistic inputs and is trained to modulate, merge and use both visual and linguistic information in order to maximize a task score. For instance, in Multi-modal Machine Translation (MMT), the model is required to translate an image description to another language.

The integration of visual input in MMT has always been the primary focus of the different researches in the field. Regional and global features have first been investigated (Huang et al., 2016), then convolutional features of higher dimensions (such as the res4f layer from ResNet) (Calixto et al., 2017; Delbrouck and Dupont, 2017b) were used because they carry more visual information. Recently, Caglayan et al. (2017) found that light architectures with fewer parameters are more suitable for the learning of the MMT task. Because of the limited number of training parameters, global features must be used. A trade-off arises : models with bigger attention mechanism could take advantage of richer visual input but the addition of training parameters seems to impair the translation quality.

To tackle this problem, we decide to take a state-of-the-art MMT model and add a conditional generator whose aim is to reconstruct the global visual input used during the translating process using only the model terminal state. We also want this reconstruction to be evaluated adversarially. This approach has four purposes :

- We constrain the model to closely represent the semantic meaning of the sentence by reconstructing the visual input. We believe it would ground the visual information into the training process and enable better generalization;
- We leave the whole translation model pipeline unchanged, no learning parameters are added for translation. The generator module is trained end-to-end during training but is unused during inference;
- Because we use light global features, the reconstruction process is very fast and require

few learning parameters;

- By using an adversarial approach, we want our generator to approximate the true data distribution of images. We believe that the propagation of generator's gradient back into the translation model would enable better generalization for unseen images on the different test-sets.

This reconstruction problem has two parts. First, we add the reconstruction module on top of our primary MMT task and investigate the different architecture for the generator. Secondly, we treat the reconstruction as an adversarial problem. We modulate our network following recent promising adversarial architectures and evaluate how the adversarial response helps the translation pipeline in its learning. We prove their efficiency by showing strong generalization the different MMT test-sets.

2 Related work

In the modality reconstruction field, the closest work related to ours is the one of (Rohrbach et al., 2016) who proposes an approach which can learn to visually localize phrases relying on phrases associated with bounding boxes in an image. Nevertheless, our works differ in two ways. First, the reconstruction is linguistic. They aim to reconstruct the sentence from a visual attention. Secondly, their visual data are annotated with bounding boxes representing linguistic information while our approach doesnt require any preprocessing.

When reconstructing its input, a model can be seen as an auto-encoder (Hinton and Salakhutdinov, 2006) which aims to compress or encode with model $Q(z|v)$ a modality v into a representation z and then decode (or reconstruct) from z an approximation v' with decoder $G(v|z)$. The difference lies in that our latent variable z (or compressed representation) is the final representation of a MMT model. Input v is modulated in by the multi-modal model before being decoded (or reconstructed). Because our latent variable z will be adversarially evaluated, our model is also close to an adversarial auto-encoders (AAE) (Makhzani et al., 2016).

Adversarial approaches for multimodal-tasks have been investigated in image-captioning (Feng et al., 2018) or visual question answering (Ilievski and Feng, 2017). In those works, the task goal is fully adversarial which differs from our approach. Our translation model is still a classification task and uses the widely adopted negative log likelihood loss. Only the reconstruction module is treated as adversarial.

Finally, reconstruction (or imagination as called in the author's paper) has been investigated with regression techniques Elliott and Kádár (2017). A major difference, besides our adversarial approaches, is their choice to not use any visual information during inference. The image is only used as training input for the reconstruction module, not the translation module. We believe it could penalize the model to do so if the information for translation really is in the image. As previously stated, using visual input for translation might impair overall translation quality but we force our model to use a visual attention during inference as it is the very foundation of the multimodal translation task.

3 Background

In this section, we describe the concepts involved in our experiments. We start by describing how visual reconstruction as an auxiliary task is built on top of our MMT model. We then explain the two adversarial settings used to involved in our experiments: a generative adversarial network and an adversarial auto-encoder.

3.1 Visual reconstruction

We denote the MMT model Q and its inputs x and v for the linguistic and visual data respectively. The model learns to output the translation y of x as formulated hereafter:

$$y, h_T = Q(x, v) \tag{1}$$

where h_T is defined as the model's Q final state (or last hidden state). A generator G takes as input h_T and approximates a visual reconstruction v':

$$v' = G(h_T) \tag{2}$$

From equation 1 and 2, we compute the total loss $\mathcal{L}_{MMT}$ of model Q and generator G :

$$\mathcal{L}_{MMT} = \overbrace{\mathcal{L}_Q(y,x)}^{\text{translation pipeline}} + \lambda_r \overbrace{\mathcal{L}_R(v',v)}^{\text{reconstruction pipeline}} \tag{3}$$

Factor λ_r indicates the weight of the reconstruction loss.

Notation used in this sub-section 3.1 are matched in the following sub-sections 3.2 and 3.3 for clarity.

3.2 Generative adversarial network (GAN)

A generative adversarial network (Goodfellow et al., 2014) is a model whose main focus is to generate new data based on source data. It is made of two networks: the generator G that constructs synthetic data from noise samples z and the discriminator D that distinguishes generated samples from the generator or from the true data-set distribution. Intuitively, one can say that the goal of the generator is to fool the discriminator by synthesizing data close to the data distribution. This leads to a competition between both networks called the min-max objective:

$$\min_G \max_D \mathbb{E}_{v\sim\mathbb{P}_{true}}[\log(D(v))] + \mathbb{E}_{v'\sim\mathbb{P}_{generated}}[\log(1-D(v'))] \tag{4}$$

where v is an example from the true data and $v' = G(z)$ a sample from the Generator and variable z is Gaussian noise.

To stabilize training and tackle the vanishing gradient problem, Gulrajani et al. (2017) introduce a gradient penalty in the objective :

$$\mathbb{E}_{v\sim\mathbb{P}_{true}}[D(v)] + \mathbb{E}_{v'\sim\mathbb{P}_{generated}}[1-D(G(z))] + \lambda_{gp}\,\mathbb{E}_{\hat{v}\sim\mathbb{P}_{\hat{v}}}[(\|\nabla_{\hat{v}}D(\hat{v})\|_2 - 1)^2] \tag{5}$$

with $\hat{v} = \epsilon v + (1-\epsilon)v'$ and where ϵ is a random number sampled from the uniform distribution $U[0,1]$ and λ_{gp} is the penalty factor. This method produces more stable gradient and the critic can match more complex distribution.

This equation refers to the wasserstein GAN (WGAN, Gulrajani et al. (2017)) with gradient penalty that will be used in our experiments at section 4.

3.3 Adversarial Auto-encoders (AAE)

Auto-encoders are made of two parts : an encoder Q receives the input v and creates a latent or hidden representation h of it, and the generator G takes this intermediate representation and tries to reconstruct the input as v'. A common loss is to use the mean square error between the input and reconstructed inputs.

$$L_R(,v') = ||v - v'||^2 \tag{6}$$

Variational autoencoders impose a constraint on how to construct the hidden representation. The encoder can not use the entire latent space freely but has to restrict the hidden codes h produced to be likely under the prior distribution $p(v)$. This can be seen as a type of regularization on the amount of information that can be stored in the latent code. The benefit of this relies on the fact that now we can use the system as a generative model. To create a new sample that comes from the data distribution $p(v)$, we sample from $p(h)$ and run this sample through the generator. In order to enforce this property a second term is added to the loss function in the form of a Kullback-Liebler (KL) divergence between the two distributions :

$$L(v,v') = L_R(v,v') + KL(Q(h|v)||p(h)) \tag{7}$$

where $Q(h|v)$ is the encoder of our network and $p(h)$ is the prior distribution imposed on the latent code.

Adversarial autoencoders (Makhzani et al., 2016) avoid using the KL divergence by using adversarial learning. In this architecture, a new discriminative network D is trained to predict whether a sample comes from the latent code of the generator $Q(h|v)$ or from the prior distribution imposed on the latent code $p(h)$. The loss of the encoder is now composed by the reconstruction loss plus the loss given by the discriminator network.

We can now use the loss incurred by the encoder of the adversarial network instead of a KL divergence for it to learn how to produce samples according to the distribution $p(h)$. The loss of the discriminator D is :

$$L_D = -\log(D(h')) + \log(1-D(h)) \tag{8}$$

where h is generated by the encoder and h' is a sample from the true prior (usually a gaussian distribution). Following the mix-max game, the loss of the encoder Q is :

$$L_Q = -\log(D(z)) \tag{9}$$

As seen in the previous sub-section, we can make this AAE wasserstein (WAAE, Tolstikhin et al. (2018)) by using the Wasserstein distance between the two probability distributions and by introducing a regularizer penalizing discrepancy between prior distribution and distribution induced by the encoder.

4 MMT Experiments

In this section, we describe the two visual reconstruction experiments on model Q evaluated in section 6.

4.1 G-WGAN

In the original algorithm, G receive z as input and is usually a sample from Gaussian noise. In the case of MMT, noise z will be concatenated with the model Q's last hidden state h_T so that the generator reconstruct the features according to the translated sentence. Generator G then becomes a conditional generative network (Mirza and Osindero, 2014) and outputs the reconstructed features $v' = G([z, h])$. This reconstruction will be evaluated by discriminator D. This settings is illustrated in figure 1. The goal of noise is to make the generator non-deterministic so that it is harder the for model D to discriminate between the real and the fake sample. Stochasticity can be induced by dropout as well (Isola et al., 2017) and will be used in our model. The full procedure can be found in Algorithm 1.

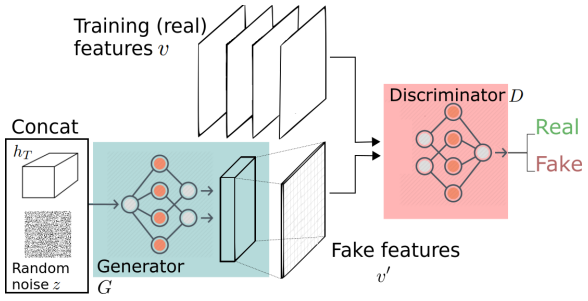

Figure 1: Training flow of G-WGAN. Model Q omitted for clarity.

4.2 Q-WAAE

In this experiment, the encoder Q is actually the multi-modal translation model Q. The latent variable h is seen as the last hidden state h_T of the model Q. D has to discriminate between the latent code h_T or the "real" latent code h' sampled from a Gaussian distribution. Along the adversarial loss, a generator G reconstruct the features v' with input h_T. The figure 2 depicts the reconstruction. The full procedure can be found in Algorithm 2.

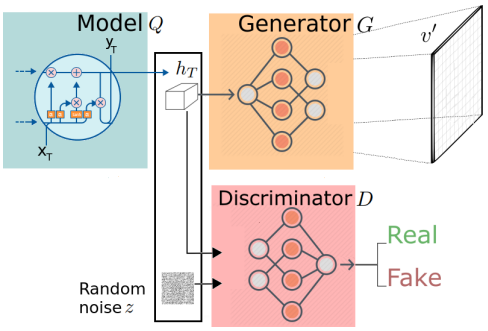

Figure 2: Training flow of Q-WAAE. The last hidden state h_T is the input for decoder P

5 Settings

In this section, we describe the model Q and the data-set used.

5.1 Training

To be consistent with the state-of-the-art, we follow the settings that are used in the previous works we compare our model to in the result section. The full description of the model Q can be found in appendix A. RNN layer size, attention size, dropout, model ensembling and training settings are left unchanged for a fair comparison.

We train jointly Q and G with Adam optimizer (Kingma and Ba, 2014) with the learning rate 4e-4 and gradient clipping is set to 1. The visual input v used are the images features from the last pooling layer (pool5) of the ResNet-50 (He et al., 2016) and are of dimension 2048×1. We use a batch-size of 32. For both task, we stop training if the task score doesn't improve for more than 5 epochs. Model reported are ensembling of 5 models.

Finally, the gradient penalty λ_{gp} is set to 10 for all experiments. For Q-WAAE, the λ_{critic} coeffi-

Algorithm 1 G-WGAN : Wasserstein GAN with gradient penality

Require: Adversarial coefficient $\lambda_a > 0$, gradient penalty coefficient $\lambda_{gp} = 10$, the number of D iterations per G iteration $\lambda_{critic} = 5$

Initialize the parameters θ of the MMT model Q, generator G and features discriminator D.
while Q not converged **do**
Sample x, v from the training set
Output translations y from $Q(x, v)$
Get last states h_T from Q
for $t = 1, ..., \lambda_{critic}$ **do**
Sample noise z from $\mathcal{N}(0, 1)$
Sample random number ϵ from $U[0, 1]$
$v' \leftarrow G([z, h_T])$
$\hat{v} \leftarrow v\epsilon + v'(1 - \epsilon)$
Update D_θ by ascending:

$$D(v) + (1 - D(v')) + \lambda_{gp} \left(\|\nabla_{\hat{v}} D(\hat{v})\|_2 - 1\right)^2$$

Update G_θ and Q_θ by descending the adversarial loss $\mathcal{L}_R$:

$$\lambda_a \; D(v')$$

Update Q_θ by descending translation loss $\mathcal{L}_Q(x, y)$

Algorithm 2 Q-WAAE : Wasserstein Auto-Encoder with gradient penalty

Require: Adversarial coefficient $\lambda_a > 0$, reconstruction coefficient $\lambda_r > 0$, gradient penalty coefficient $\lambda_{gp} = 10$

Initialize the parameters θ of the MMT model Q, generator G and latent discriminator D.
Use mean square error as c.
while Q not converged **do**
Sample x, v from the training set
Output translations y from $Q(x, v)$
Get last states h_T from Q
Sample "true" state h' from $\mathcal{N}(0, 1)$
Sample random number ϵ from $U[0, 1]$
$\hat{h} \leftarrow h_T\epsilon + h'(1 - \epsilon)$
Update D_θ by ascending:

$$D(h') + (1 - D(h_T)) + \lambda_{gp} \left(\|\nabla_{\hat{h}} D(\hat{h})\|_2 - 1\right)^2$$

Update G_θ and Q_θ by descending reconstruction and adversarial loss $\mathcal{L}_R$:

$$\lambda_r \; c(v, G(h_T)) - \lambda_a \; \log(D(h_T))$$

Update Q_θ by descending translation loss $\mathcal{L}_Q(x, y)$

cient is set to 5. The adversarial and reconstruction coefficients λ_a and λ_r are detailed in the results section 6. The discriminator D is trained with adam with learning rate of 2e-4, β_1 = 0.5 and β_2 = 0.9. The architecture of G and D is available in Appendix B. We found out that the use spectral normalization (Miyato et al., 2018) and batch normalization didn't improve the translation scores.

5.2 Dataset

We use the Multi30K dataset (Elliott et al., 2016). For each image, one of the English descriptions was selected and manually translated into German by a professional translator. As training and development data, 29,000 and 1,014 triples are used respectively. We use the three available test sets to score our models. The Flickr Test2016 and the Flickr Test2017 set contain 1000 image-caption pairs and the ambiguous MSCOCO test set 461 pairs. Recently, a fourth dataset, the Flickr Test2018 set, is used for the online competition on codalab [1]. It consists of 1,071 sentences is released without the German and French gold translations.

6 Results

We now report the results for the different two configurations introduced in section 4 on the Multi-modal Machine Translation (MMT) task. All experiments reported were run on a single NVIDIA GTX 1080 GPU.

6.1 Quantity evaluation

First and foremost, we notice that the most successful model is Q-WAAE as it marginally surpasses the baseline and previous works in every dataset. It is also the best official reported score as constrained submission (only data provided by the challenge) of the test 2018 data-set. The

[1] https://competitions.codalab.org/competitions/19917#results

Test sets	Test 2016 Flickr		Test 2017 Flickr	
	BLEU	**METEOR**	**BLEU**	**METEOR**
FAA(2018)	-	-	31.60	52.50
DeepGru(2018)	40.34	59.58	32.57	53.60
Baseline	40.00	59.20	32.20	53.10
G-WGAN	40.38 +0.38	60.03 +0.83	33.70 +1.50	54.50 +1.40
Q-WAAE	40.66 +0.66	60.06 +0.86	34.06 +1.86	54.94 +1.84
Test sets	**COCO-ambiguous**		**Test 2018 Flickr**	
FAA(2018)	-	-	31.39	51.43
DeepGru(2018)	29.21	49.45	31.10	51.64
Baseline	28.50	48.80	-	-
G-WGAN	31.08 +2.58	50.43 +1.63	31.80	52.15
Q-WAAE	31.41 +2.91	50.95 +2.15	**31.91**	**52.37**

Table 1: Results on the en→de MMT task. Test 2018 results (anonymized) can be checked on the official leaderboard (https://competitions.codalab.org/competitions/19917#results) in the "german" tab. Score differences are computed against the baseline.

submission surpasses the previous best METEOR score from DeepGru by 0.73 METEOR and the previous best BLEU score from FAA by 0.52 points. More importantly, the Q-WAAE model significantly improves the SOTA on the COCO-ambiguous data-set, a test-set that has been specifically designed to include 56 unique ambiguous verbs in 461 descriptions ($+2.91$ BLEU and $+1.63$ METEOR).

		λ_r		
		0.2	0.5	0.8
λ_a	0.2	**50.95**	50.08	49.33
	0.5	49.79	49.62	49.16
	0.8	49.70	49.16	48.02

Table 2: Q-WAAE : Impact on the METEOR metric of the reconstruction and adversarial loss coefficient on the ambiguous COCO data-set

To try and get the best results on the Q-WAAE, we mixed different combinations of the coefficient factors on the adversarial and reconstruction loss as shown in table 2. The results show that if the auxiliary loss (adversarial and/or reconstruction) is made too important compared to the translation loss, the translation quality is impaired.

The G-WGAN also shows improvements over the baseline and obtains similar results to Q-WAAE. Nonetheless, a small discrepancy is noticeable on the COCO-ambiguous. We believe that the main advantage of the Q-WAAE loss is the actual presence of a direct mean square error reconstruction loss along the adversarial loss. We also noticed that the G-WGAN model is really sensitive to the dimension of noise concatenated to the hidden state given as input to the generator as stated in table 3.

	$\lvert z\rvert$			
	64	128	256	512
METEOR	50.35	**50.43**	49.71	49.48

Table 3: G-GWAN : Impact of the noise concatenated to the hidden state of size 512

One can argue that because the generator is conditional on the hidden state h_T which is of high dimension, its very hard for the generator to become deterministic. An important noise dimension could potentially harm the generator instead of fooling the discriminator.

6.2 Quality evaluation

To understand the success of Q-WAAE on the ambiguous COCO data-set, we perform an ablation study of the model. We first discard the adversarial discriminator so that we only train the reconstruction module with the MSE loss (+ G). We also discard the use of the features v in the translation model for both the ablated model and Q-WAAE (no v). The results of the ablation study can be found in table 4.

Test sets	COCO-ambiguous	
	BLEU	**METEOR**
Baseline	28.50	48.80
Baseline + G + no v	29.43	49.60
Baseline + G	29.91	49.24
Q-WAAE + no v	30.57	50.15
Q-WAAE	31.41	50.95

Table 4: Ablation study of Q-WAAE model

A first observation is that the reconstruction module G does improve the baseline, but the the Baseline + G + no v model (no the visual input in the translation pipeline) has a better METEOR metric than the Baseline + G model. It means that use of a visual attention model in the translation pipeline harms the overall translation quality, as already found in previous work. In contrast, Q-WAAE hopefully performs better than Q-WAAE + no v, which shows the successful integration of the visual input, as it should be expect for the MMT task. Using adversarial feedback does provide a stronger training and a better generalization over the different data-sets.

6.3 Improvements examples

To further investigate the quality of the Q-WAAE model, we pick two examples to illustrate the improvements.

In figure 3, the baseline translates "pointing a camera" to "zeigt auf ein camera" which could translate to "to point at a camera". It is incorrect since the image displays the camera-man pointing *a* camera *at* the speaker. Also, the german verb "zeigen" also means to show, to demonstrate, which is not ideal in this example. Our model

En: a cameraperson pointing a camera and mic at a speaker .
GT: eine kamerakraft richtet eine kamera und ein mikrofon auf einen sprecher .
Baseline: eine kameraperson zeigt auf eine kamera und mikrofon an einem redner .
Q-WAAE: eine kameraperson richtet eine kamera und ein mikrofon an einem redner .

Figure 3: An ambiguous COCO example where Q-WAAE finds the right translation for the verb

translates "pointing" to "richtet" meaning "pointing" with the idea of aiming which is more suitable. Also Q-WAAE does not use wrong prepositions. The sentence of baseline scores a BLEU of 0 while the sentence score of our model is a BLEU of 44.83.

En: a woman winding up to pitch a softball .
GT: eine frau bereitet sich darauf vor , einen softball zu werfen .
Baseline: eine frau bereitet sich auf einen softballwurf vor .
Q-WAAE: eine frau bereitet sich darauf vor , einen softball zu werfen .

Figure 4: An ambiguous COCO example where Q-WAAE finds the right translation for the object

The second figure aims to show that not only Q-WAAE manages to correctly translates ambiguous verbs but more complex examples. In Figure 4, the Q-WAAE model ends up getting the perfect translation (a BLEU score of 100) whereas the baseline model outputs a translation closer to "a woman winding up for softball", missing the second verb (BLEU score of 22.60).

6.4 Other data-set

We decided to train Q-WAAE on another language pair of the Multi30K dataset, namely the en $\rightarrow$ fr pair. Again the model surpasses the baseline for the COCO-ambiguous and test 2018 test sets.

	BLEU	METEOR
Test sets en → fr	**COCO-ambiguous**	
DeepGru	46.16	65.79
Q-WAAE	47.00	66.50
	Test 2017	
DeepGru	55.13	71.52
FAA	52.80	69.60
Q-WAAE	56.54	72.32
	Test 2018	
FAA	39.48	59.85
Q-WAAE	**40.09**	**60.54**

Table 5: Results on the en → fr Multi30K dataset, test 2018 results can found online in the aforementioned codalab link in the "french" tab

7 Conclusion

We demonstrated that recent advances in adversarial generative modeling was able to successfully ground visual information for multi-modal translation using visual and linguistic input. We show that the use of visual information for the model still remains a challenging task. The presented work in this paper aimed to modulate the last hidden state at the end of the translation model, it would be interesting to investigate adversarial approaches more upstream in the pipeline like in the visual features extraction (as previously investigated in (Delbrouck and Dupont, 2017a)).

References

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost Van de Weijer. 2017. Lium-cvc submissions for wmt17 multimodal translation task. *arXiv preprint arXiv:1707.04481*.

Ozan Caglayan, Adrien Bardet, Fethi Bougares, Loc 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, Volume 2: Shared Task Papers*, pages 603–608, Belgium, Brussels. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. 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)*, pages 1913–1924.

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

Jean-Benoit Delbrouck and Stephane Dupont. 2017b. Multimodal compact bilinear pooling for multimodal neural machine translation. *arXiv preprint arXiv:1703.08084*.

Jean-Benoit Delbrouck and Stéphane Dupont. 2018. Umons submission for wmt18 multimodal translation task. In *Proceedings of the First Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

D. Elliott, S. Frank, K. Sima'an, and L. Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

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

Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. 2018. Unsupervised image captioning. *arXiv preprint arXiv:1811.10787*.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. 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*, pages 5767–5777. Curran Associates, Inc.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, 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.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

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: Volume 2, Shared Task Papers*, pages 639–645.

Ilija Ilievski and Jiashi Feng. 2017. Generative attention model with adversarial self-learning for visual question answering. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 415–423. ACM.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137. IEEE Computer Society.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. 2016. Adversarial autoencoders. In *International Conference on Learning Representations*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer.

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)*, pages 1715–1725. Association for Computational Linguistics.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Scholkopf. 2018. Wasserstein auto-encoders.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

A Model Q

Given a source sentence $\boldsymbol{x}$ and visual features $\boldsymbol{v}$, an attention-based encoder-decoder model outputs the translated sentence $\boldsymbol{y}$. If we denote $\boldsymbol{\theta}$ as the model parameters, then $\boldsymbol{\theta}$ is learned by maximizing the likelihood of the observed sequence $\boldsymbol{y}$ or in other words by minimizing the cross entropy loss. The objective function is given by:

$$\mathcal{L}_{\mathcal{Q}}(\boldsymbol{\theta}) = -\sum_{t=1}^{n} \log p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{y}_{<t}, \boldsymbol{v}, \boldsymbol{x}) \quad (10)$$

Three main components are involved: an encoder, a decoder and an attention model.

Encoder The encoder is a bidirectional-GRU that create a set of annotation S:

$$\boldsymbol{S} = \begin{bmatrix} \text{GRU}_{\text{forward}}(\overrightarrow{\boldsymbol{x}}) \\ \text{GRU}_{\text{backward}}(\overrightarrow{\boldsymbol{x}}) \end{bmatrix}$$

A word x_t has an embedding of 256, each GRU is of size 512 thus annotation $\boldsymbol{S}$ are of size 1024.

Decoder The decoder is a conditional GRU (cGRU). The following equations describes a cGRU cell :

$$\begin{aligned} \boldsymbol{h}'_t &= \text{GRU}_1(y_t, \boldsymbol{h}_{t-1}) \\ \boldsymbol{c}_t &= \text{ATT}(\boldsymbol{h}'_t, \boldsymbol{v}, \boldsymbol{S}) \\ \boldsymbol{h}_t &= \text{GRU}_1(\boldsymbol{h}'_t, \boldsymbol{c}_t) \end{aligned} \quad (11)$$

where both GRU have 512 units and ATT is the attention module defined hereafter :

$$\boldsymbol{a}'_t = \boldsymbol{W}^a \tanh(\boldsymbol{W}^{\text{h}}\boldsymbol{h}'_t + \boldsymbol{W}^{\text{s}}\boldsymbol{S}) \quad (12)$$

$$\boldsymbol{a}_t = \text{softmax}(\boldsymbol{a}'_t) \quad (13)$$

$$\boldsymbol{c}'_t = \sum_{i=0}^{M-1} \boldsymbol{a}_{t_i}\boldsymbol{s}_i \quad (14)$$

$$\boldsymbol{i}_t = \tanh(\boldsymbol{W}^{\text{feat}}\boldsymbol{v}) \quad (15)$$

$$\boldsymbol{c}_t = \boldsymbol{W}^c(\boldsymbol{c}'_t \odot \boldsymbol{i}_t) \quad (16)$$

Matrices $\boldsymbol{W}^s$ and $\boldsymbol{W}^h$ map respective inputs to size 1024 $\boldsymbol{W}^h$. $\boldsymbol{W}^{\text{feat}}$ transform visual features to size 1024 and $\boldsymbol{W}^c$ transforms both attention vector back to size 512 to be compatible with GRU_2 size.

Finally, a bottleneck function projects the cGRU output into probabilities over the target vocabulary. It is defined so:

$$\boldsymbol{b}_t = \tanh(\boldsymbol{W}^{\text{bot}}\boldsymbol{h}_t) \quad (17)$$

$$y_t \sim \boldsymbol{p}_t = \text{softmax}(\boldsymbol{W}^{\text{proj}}\boldsymbol{b}_t) \quad (18)$$

where $\boldsymbol{W}^{\text{bot}}$ maps hidden state to size 256 and $\boldsymbol{W}^{\text{proj}}$ maps the bottleneck result to the vocabulary size.

Dropout of 0.3 is used on embeddings $\boldsymbol{x}$ and annotations $\boldsymbol{S}$ and of 0.5 on $\boldsymbol{b}_t$.

To marginally reduce our vocabulary size, we use the byte pair encoding (BPE) algorithm on the train set to convert space-separated tokens into sub-words (Sennrich et al., 2016). With 10K merge operations, the resulting vocabulary sizes of each language pair are: 5204 $\rightarrow$ 7067 tokens for English$\rightarrow$ German and 5835$\rightarrow$ 6577 tokens for English$\rightarrow$French.

B Generator G and discriminator D

Q-WAAE Generator G is defined as follows:

$$\boldsymbol{v}' = \tanh(\boldsymbol{W}^{\text{rec}}h_T)$$

where $\boldsymbol{W}^{\text{rec}}$ is of size 512×2048.

Discriminator D is defined as follows :

$$\boldsymbol{o} = \boldsymbol{W}^{\text{adv}}h_T$$

where $\boldsymbol{W}^{\text{adv}}$ is of size 512×1.

G-WGAN Generator G is defined as follows:

$$\boldsymbol{v}' = \tanh(\boldsymbol{W}^{\text{rec}}[z, h_T])$$

where $\boldsymbol{W}^{\text{rec}}$ is of size 640×2048.

Discriminator D is defined as follows ($\boldsymbol{v}$ is either real $\boldsymbol{v}$ or generated $\boldsymbol{v}'$):

$$\boldsymbol{o}_1 = \text{relu}(\boldsymbol{W}^{\text{adv}_1}[\boldsymbol{v}, h_T]) \quad (19)$$

$$\boldsymbol{o}_2 = \text{relu}(\boldsymbol{W}^{\text{adv}_2}\boldsymbol{o}_1) \quad (20)$$

$$\boldsymbol{o}_3 = \boldsymbol{W}^{\text{adv}_3}\boldsymbol{o}_2 \quad (21)$$

where $\boldsymbol{W}^{\text{adv}_1}$ is of size 2560×1024, $\boldsymbol{W}^{\text{adv}_2}$ of size is of size 1024×512 and $\boldsymbol{W}^{\text{adv}_3}$ of size is of size 512×1