Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Probing the Role of Video Context in Multimodal Machine Translation with Source Corruption

Author:
Zixiu Wu

Supervisor:
Prof. Lucia Specia

Submitted in partial fulfillment of the requirements for the MSc degree in Advanced Computing of Imperial College London

September 2019

Abstract

Multimodal machine translation (MMT) is the task of utilising information from non-textual modalities to aid textual machine translation (MT). Thanks to the recent rapid development of deep neural networks (DNNs) and their application in neural machine translation (NMT) and computer vision, the past few years have witnessed booming research in neural MMT that seeks to achieve visually grounded NMT based on visual features from DNNs. However, the question still remains as to if and how visual information helps MMT models translate. Previous work has explored source degradation to investigate the impact of image features on MMT model performance. However, little has been done for the video domain, where the visual modality contains considerably richer information. We therefore carry out a two-phased project focused on video-based MMT with source corruption. In Phase 1, we train multimodal transformers on video subtitles with different degrees of verb masking, and then conduct incongruence and human analyses to assess the importance of visual information. In Phase 2, multimodal transformers and deliberation networks are trained for multimodal cascaded speech translation (MCST) before their attention is visualised and their normal and incongruent performance is assessed on varying levels of noisy transcripts. Our Phase 1 results show that the multimodal transformers deliver competitive performance while improving translation quality on verb-masked source sentences. Our Phase 2 results, however, reveal that multimodal models largely fail to utilise visual information to bridge the semantic gap between the transcripts and original subtitles. We also discover in Phase 2 that multihead visual attention works poorly on convolutional visual features and comparatively better on action category embeddings, but only when the source sentence is strongly related to the video.

Acknowledgments

I have been fortunate enough to have received substantial help throughout the project. I would like to first thank my advisor, Prof. Lucia Specia, who, with her expertise in machine translation and natural language processing in general, has been offering invaluable advice on my research topics and methodology ever since my Independent Study Option. I would also like to thank Dr. Julia Ive for her helpful suggestions when I was confused or stuck. Last but by no means the least, I want to thank Dr. Ozan Caglayan, Dr. Josiah Wang and Dr. Pranava Madhyastha for the resources and materials for the project, without whom this thesis would not have been possible.

Contents

Chapter 1

Introduction

As arguably the most important channel through which humans convey their thoughts and feelings, language has always drawn enormous research interest. Among the numerous language-related fields, translation is perhaps one of the oldest, as it brings convenience to the communication between people speaking different languages and makes foreign cultures more accessible. However, translation per se is a very resource-consuming and often tedious task, therefore its automation — machine translation (MT) (Weaver, 1955)— has been studied for decades.

Once dependent on specialised linguistics expertise to develop rules, MT has evolved greatly and is not at all language-pair-dependent nowadays. This shift was thanks to the advent of statistical machine translation (SMT) (Brown et al., 1988) and, more recently, neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013). The former achieved probabilistic modelling of languages while the latter has successfully for the most part removed the need for human-designed feature engineering and delegated virtually everything to neural-network(McCulloch and Pitts, 1943)-based models. Along with the dramatically increased automation has been the significant improvement of translation quality—nowadays the overwhelming majority of MT applications and research are driven by neural networks.

The inevitable limitation of NMT, however, is its largely unimodal nature. Despite major architectural breakthroughs such as the attention mechanism (Bahdanau et al., 2014) and the transformer (Vaswani et al., 2017), NMT still only sees the source text during translation. Translation these days is more often than not more than text — video subtitling, for instance, is an inherently multimodal task. It is extremely unlikely that the professional translators of a movie would produce the subtitles by inspecting the original-language subtitles alone, not to mention that it is rare in real life to have the subtitles of a random video ready for translation.

Hence the birth of multimodal machine translation (MMT) (Elliott et al., 2015; Hitschler et al., 2016). The benefits of having non-textual-modality information are obvious. Disambiguation, for instance, is considerably easier with visually grounded translation. Given the sentence “The pianist has finished the sonata” alone, an NMT model has no way to know for sure whether to translate “the pianist” into “le pianiste" (masculine) or "la pianiste" (feminine) in French since the gender of the pianist cannot be deduced. In fact, at the time of writing, the translation that Google Translate produces is "le pianiste", very possibly due to the gender bias in the training data (the same goes for words such as "player"). If the sentence is the subtitle of a video featuring a female pianist, however, a well-trained MMT model will easily choose to translate it as "la pianiste", just as a human would.

In recent years, MMT has largely aimed at combining the textual and visual (image) modalities, mostly due to the rapid progress in computer vision (CV) that has enabled sophisticated visual feature extraction. Well-known convolutional neural networks (CNNs) (LeCun et al., 1989) such as ResNet (He et al., 2016) and VGG-16 and 19 (Simonyan and Zisserman, 2014) have been adopted to capture the semantics of visual information for MMT. Therefore, MMT research has in essence evolved into various ways of integrating visual features into NMT models to complement the source sentence and thus achieve higher-quality translation.

In MMT research, the visual features often come from the fully-connected layer or the softmax layer as a single global feature vector or from the last convolutional layer as a grid of image region representations. On datasets such as Multi30K (Elliott et al., 2016), those multimodal models show better translation quality than the text-only baseline, and provide numerous examples where visual information is indeed utilised to aid translation, such as where the model attends to relevant image regions for translating a related word.

However, the contribution of the visual modality to translation quality in the MMT models proposed so far is still an open question. For instance, the organisers of the MMT shared task have the opinion (Barrault et al., 2018a) that the MMT models devised heretofore have led to mostly insignificant decoding differences according to automatic metrics and human judgement, while Elliott (2018) observes no serious performance degradation when MMT models are fed with randomly assigned image features instead of the correct ones.

To probe this matter further, Caglayan et al. (2019a) carry out a number input degradation schemes on Multi30K, including masking colour words and visually present entities, and find that visual information boosts translation quality the most when the corrupted source sentence is inadequate for correct decoding. When the source sentence is sufficient, however, the sensitivity to the visual modality drops.

Inspired by Caglayan et al. (2019a), we, too, explore the effect of the visual modality in the presence of source corruption, but this time in the video domain, based on the assumption that videos generally contain richer information than images and therefore could offer more to MMT. We conduct our experiments on How2 (Sanabria et al., 2018), a dataset of instructional videos on YouTube with English subtitles and Portuguese translations.

Specifically, in two phases, we investigate two separate scenarios: artificial corruption and random noise. For the former, we execute different degrees of verb masking on the English subtitles before training multimodal transformers to translate those corrupted subtitles into the correct Portuguese translations, as we expect the visual features from the videos to compensate for the source corruption. For the random noise scenario, we use a monomodal ASR model to transcribe the videos and then train multimodal transformers and deliberation networks (Xia et al., 2017) to translate those transcripts into the correct Portuguese translations. In doing so, we effectively achieve multimodal cascaded speech translation (MCST) as well as introducing random noise to the source via transcribing.

Our main experiments include the following:

- training multimodal transformers to translate How2 English subtitles with different extents of verb masking (Phase 1)
- incongruent analysis and human analysis for the multimodal transformers above for more insight into the models and results (Phase 1)
- training multimodal transformers and two types of deliberation networks for MCST (Phase 2)
- analysing MCST performance with a novel “incongruence + transcript faithfulness” test as well as multihead attention visualisation (Phase 2)

In this thesis, Chapter 2 visits the basics of NMT, the building blocks relevant to our experiments, as well as recent MMT models. We introduce the design of our systems in Chapter 3, including the speech recognition system, our visual features, and the transformer and deliberation networks. The details of our experiments are given in Chapter 4 and we show the results and analyses in Chapter 5. Finally, Chapter 6 concludes the thesis by listing our contributions and pointing at directions for future work.

Chapter 2

Background

This project is research about multimodal machine translation (MMT) (Elliott et al., 2015; Hitschler et al., 2016), a booming area of natural language processing (NLP) connecting the visual and textual domains. The field is relatively new and based upon machine translation (MT) (Weaver, 1955), in particular neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013). Therefore, this chapter serves as an introduction to the above, with Chapter 2.1 offering a peek at the development of MT, Chapter 2.2 detailing the NMT building blocks that are essential to MMT and this project, and Chapter 2.3 visiting relevant MMT research.

2.1 Timeline of Machine Translation Development

Among all the sub-fields of natural language processing (NLP), machine translation (MT) (Weaver, 1955) is arguably one of the oldest. Defined simply as translating a source-language sentence into a target language, it is still one of the most thriving NLP research areas.

At the beginning, MT was approached by linguists in the middle of the 20^{th} century based on sets of human-defined rules. In those times, a source sentence would first go through grammatical analysis, its words would be mapped to the target-language domain in accordance with a pre-defined dictionary, and finally a candidate translation would be produced utilising the outcome of the previous steps in a strictly rule-based manner. Unsurprisingly, such procedures were burdensome to design, thanks to the complexities of languages. This approach was also financially demanding, and its efficacy was not guaranteed — often unsatisfactory when applied in uncontrolled environments.

Statistical machine translation (SMT) (Brown et al., 1988), delegating many MT responsibilities to probability and statistics, started to gain traction at the end of the last century, and was in a dominant position until the early 2010s. The goal is simple: maximise the posterior probability of a candidate translation being the "correct" translation of a source sentence. As its name suggests, the "correctness" in SMT depends on the statistics, i.e. the source-target language corpora. Through probabilistic modelling of those corpora, the "correctness" probability can be computed. Despite the less manual nature of SMT, it still requires hand-crafted feature engineering, adding to the difficulty of the models as well as their dependence on specific languages and language pairs.

For the past few years, neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013) has become the go-to approach in most MT tasks. With neural networks (McCulloch and Pitts, 1943) to learn features automatically, the human has a much smaller role to play, making the models a lot simpler as well as more efficient. Sequence-to-sequence (seq2seq) (Sutskever et al., 2014) is the predominant type of NMT models, where an encoder learns to map a source-language sentence into a high-dimensional semantic space and a decoder on the other hand learns to map the representation back into the domain of target-language sentences. Benefits of this approach include intuitive trainability on corpora, completely language-independent feature learning and, most importantly, greatly improved performance over SMT.

2.2 Neural Machine Translation Architectures

2.2.1 Vanilla RNNs for Seq2Seq MT

With the underlying seq2seq structure, what makes NMT models distinct from each other is how the encoder is set up to represent sentences as tensors in the high dimensional semantic space and how the decoder learns to do the reverse.

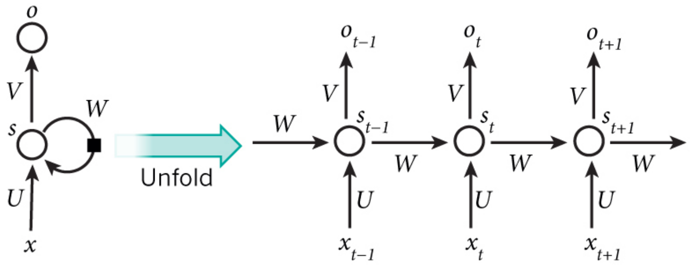


Figure 2.1: Vanilla RNN. Source: LeCun et al. (2015)

A recurrent neural network (RNN) (Rumelhart et al., 1988; Werbos et al., 1990), as shown in Figure 2.1, is an intuitive choice for both the encoder and the decoder, and indeed has been popular among researchers. Its modus operandi is to unfold temporally, using a hidden state (s, in this example) in the high dimensional semantic space to represent the semantics of the sentence it has processed thus far. In the simplest form of an RNN, the hidden state s is initialised in some manner (s_0), and then the model reads as input a new word x_t at time step t as well as the hidden state value s_{t-1} from the preceding time step $t-1$. With an activation function ($tanh$, for instance. Not shown in this example) and matrix multiplication on the inputs (U for x_t, W for s_{t-1}), the network produces a new hidden state s_t for the current time step t. An output o_t may also be generated from such a process as a result .

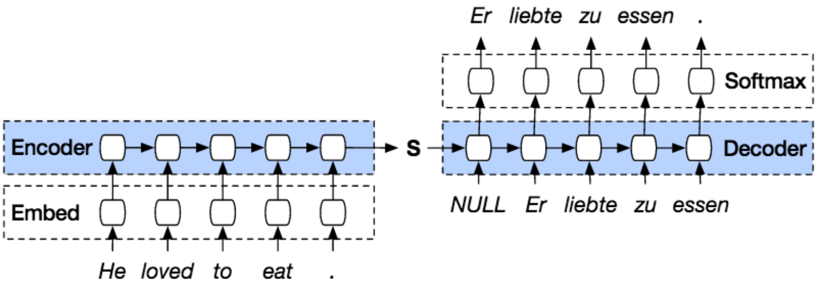


Figure 2.2: RNN-Based Seq2Seq MT. Source: Merity (2016)

Figure2.2 shows an example of an RNN-based seq2seq MT model. The encoder simply reads the source sentence word by word as embeddings and carries out the aforementioned hidden-state procedure. The decoder is, however, autoregressive, in that the word it reads at each time step as input is in fact its output from the previous time step. Also depicted in Figure 2.2 is the common practice: initialising the decoder with the final hidden state of the encoder. The translation of this model is thus obtained by converting the word-by-word output of the decoder from the embedding space to the target-language domain, in this case with a softmax function (Bridle, 1990).

2.2.2 Powerful RNN Variants

For most RNN-based MT models, the architecture relies on powerful variants of RNNs.

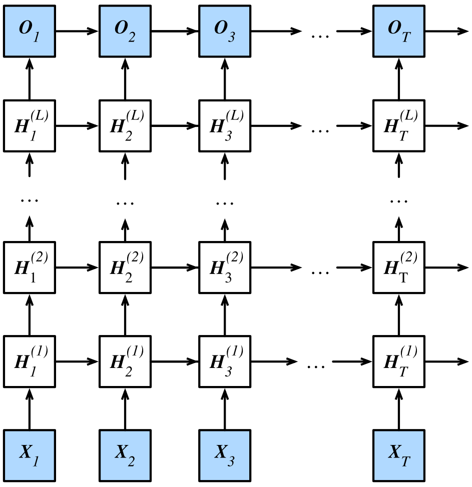


Figure 2.3: Multi-layer RNN. Source: Zhang et al. (2019)

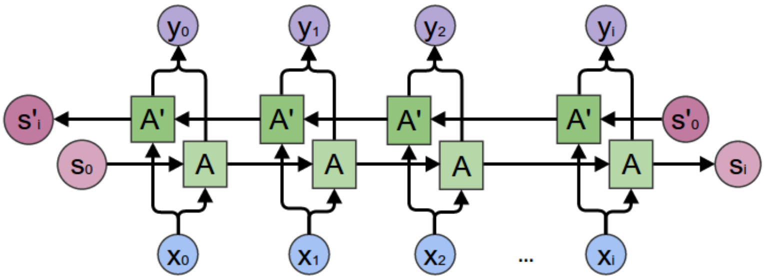


Figure 2.4: Bidirectional RNN. Source: Olah (2015a)

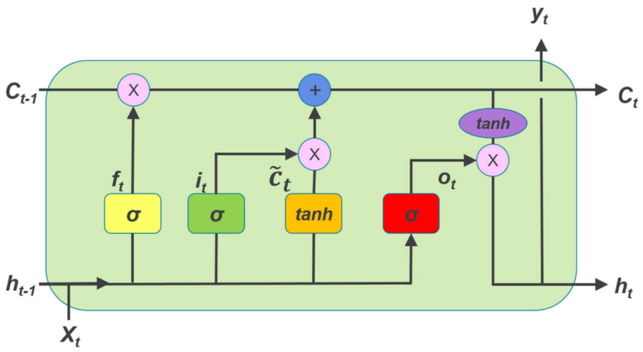
Multi-layering (Tutschku, 1995), for instance, is a straightforward yet effective structural change for RNNs. A multi-layer RNN, as shown in Figure 2.3, simply maintains a hierarchical hidden state system, so that the inputs to H_t^l, the hidden state at the l^{th} layer at time step t, are H_t^{l-1} and H_{t-1}^l, respectively the previous-time-step hidden state at the same layer and the current-time-step hidden state from the layer below (which is an input sentence word itself when $l = 1$). The output at each time step is thus generated by the top-layer hidden state. With such a hierarchical architecture, deeper learning of the input is possible.

Another tweak is bidirectionality (Schuster and Paliwal, 1997), its simplest form given in Figure 2.4. The idea addresses the forward, first-word-to-last nature of a vanilla RNN reading its input — each hidden state only takes into account the chronologically past input (x_0, x_1, x_t) words as its context, ignoring the potentially helpful information from the future (x_{t+1}, x_{t+2}, x_i). Therefore, a bidirectional RNN maintains a pair of hidden states: one of them forward (A) just as the vanilla RNN while the other (A') running in reverse direction. The concatenation of A_t and A_t' thus becomes the hidden state of the network at time step t, and this approach has proved effective in tackling the aforementioned only-see-the-past issue.

A major weakness of the vanilla and their variants above is unsatisfactory performance at learning long-term dependencies, and another is the vanishing gradient problem — due to the repeated matrix multiplication (i.e. the W in Figure 2.1) of the hidden state, it becomes difficult for backpropagation through time (BPTT) (Werbos et al., 1990) to the early time steps to be effective, as the magnitude of the gradient flow shrinks rapidly.

To alleviate the two problems above, long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) as well as gated recurrent units (GRUs) (Chung et al., 2014) were developed and have been widely applied.

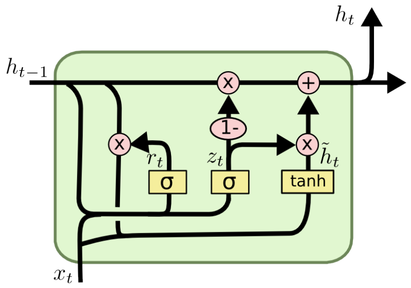
The core idea of the LSTM is to have control over the importance of the previous-step hidden state and of the current-step input to the current-step hidden state. As shown in Figure 2.5, the LSTM maintains a protected and controlled cell state C_t at each time step. Since there is no matrix multiplication involved in going from C_{t-1} to C_t, this forms an "information highway" and remedies the vanishing gradient problem. Also, i_t, f_t and o_t are the input gate, forget gate, and output gate respectively, computed based on x_t (current-step input word) and h_{t-1} (previous-step



$$
\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i\right)\\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f\right)\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o\right)\\
\tilde{\mathbf{c}}_t &= \tanh\left(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right)\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t\\
\mathbf{h}_t &= \tanh\left(\mathbf{o}_t \odot \mathbf{c}_t\right)
\end{aligned}
$$

Figure 2.5: Long Short Term Memory. Source: Ismail et al. (2018)

hidden state). With those learnable gates, the model is able to learn to control how much of the past (previous-step cell state, C_{t-1}) to forget, how much of the new information (or "candidate new cell state", $\widetilde{C}_t$) to write into the current-step cell state (C_t), and how much of the current-step cell state to write to the current-step hidden state (h_t). Thanks to this level of proactive information flow control, the LSTM has proved successful in capturing long-term dependencies.



$$
\begin{aligned}
z_t &= \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)\\
r_t &= \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)\\
\tilde{h}_t &= \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)\\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$

Figure 2.6: Gated Recurrent Unit. Source: Olah (2015b)

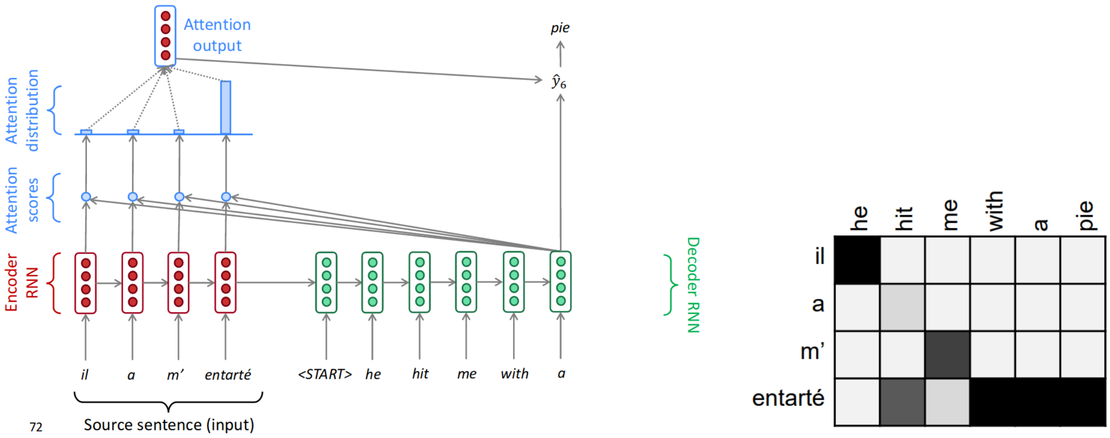
The GRU (Figure 2.6) on the other hand is a simplification of the LSTM. It coalesces the cell state C_t and the hidden state h_t of the LSTM into a single GRU hidden state h_t, and uses a single update gate z_t to replace the input gate i_t and the forget gate f_t in the LSTM. It keeps a reset gate r_t to control how much the previous hidden state h_{t-1} matters to computing the "candidate new hidden state" $\widetilde{h}_t$. Despite its reduced complexity, the GRU has been found effective in a large number of scenarios and have been popular.

An RNN-based seq2seq model these days is often a combination of the tricks above, a typical example of which is bidirectional multi-layer GRU or LSTM. They, along with other variants, cement the position of the RNN as the backbone of many NMT systems.

2.2.3 Attention

Powerful as the aforementioned RNN architectures may be, they do not address a key bottleneck for MT systems: the connection between the encoder and the decoder. As is depicted in Figure 2.2, the only connection between the encoder and decoder RNNs is the final hidden state of the former being used to initialise the latter. This practice in essence forces the encoder to summarise the input sentence as the final hidden state, and the decoder is only allowed to decipher based on that.

The attention mechanism (Bahdanau et al., 2014) was proposed precisely to remedy this issue. Its basic idea is to offer context for the decoder when the latter is generating the next word. An example is shown in Figure 2.7a, where the decoder translates the French sentence "il a m'entarté" into English: he hit me with a pie. During decoding, the decoder should of course utilise information from the previous hidden state and output, but it is also helpful to have access to the context: to generate the word "he", the decoder should pay adequate attention to the French counterpart of the word, i.e. "il", in the source sentence.



(a) Attention Computation **(b)** Attention Visualisation

Figure 2.7: Attention Computation & Visualisation. Source: See (2019)

Hence the gist of the attention mechanism: quantify the amount of attention that the decoder ought to pay to every word in the input sentence, and use that information, together with the decoder hidden state and output from the previous time step, to generate the next word.

To implement the mechanism, the encoder hidden state h_i^e of the i^{th} source word s_i of an L-word input sentence, as well the conventional decoder hidden state h_t^d at the t^{th} time step, are operands of a dot product operation that yields a raw score $\alpha_{t,i}$. With a softmax function based on all the raw scores at the same time step, i.e. $\{\alpha_{t,j} \mid 1 \leq j \leq L\}$, a normalised score $\alpha'_{t,i}$ is produced to represent the amount of attention the decoder should pay to the i^{th} source word at the t^{th} decoding step.

$$
\begin{aligned}
h_i^e &= encoder\_function(h_{i-1}^e, s_i) \\
h_t^d &= decoder\_function(h_{t-1}^d, o_{t-1}) \\
\alpha_{t,i} &= (h_i^e)^T \cdot h_t^d \\
\alpha_{t,i}^{'} &= \frac{e^{\alpha_{t,i}}}{\sum_{j=1}^{L} e^{\alpha_{t,j}}} \\
\hat{h_t^d} &= \sum_{k=1}^{L} \alpha_{t,k}^{'} \; h_k^e \\
o_t &= feedforward\_neural\_network(h_t^d, \hat{h_t^d})
\end{aligned}
\tag{2.1}
$$

The normalised scores $\{\alpha_{t,j}^{'} \mid 1 \leq j \leq L\}$ are used as weights to calculate the weighted sum $\hat{h_t^d}$ of all the encoder hidden states $\{h_k^e \mid 1 \leq k \leq L\}$ as the attention information w.r.t. the whole input sentence at the t^{th} decoding step. $\hat{h_t^d}$, commonly referred to as the context vector, along with the conventional decoder hidden state h_t^d, are fed to a feedforward neural network (Rumelhart et al., 1988) that produces the final output o_t for the current time step. See Equations 2.1 for the mathematical formulation of the whole process.

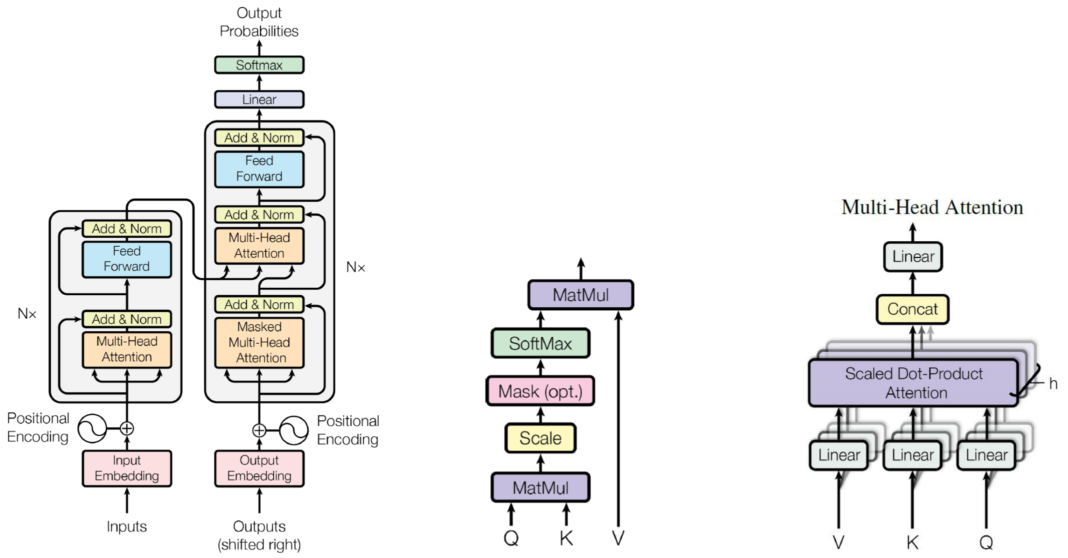
Figure 2.7b shows the attention scores after this procedure. As expected, the decoder attends considerably to “il” when generating “he”, and the same applies to “m’” – “me” as well as “entarté” – “with a pie”.

By remedying the bottleneck with attention, the mechanism enables significant improvement of translation quality and interpretability, and has become the pillar of many modern NMT models.

2.2.4 Transformer

The Transformer (Vaswani et al., 2017) pushes attention even further — it bases the hidden state representation on attention entirely.

The structure starts with its unique self-attention scheme, as shown in Figure 2.8b. Specifically, for an input sequence at layer l: $[s_1^l, s_2^l, \cdots, s_N^l]$, three separate hidden states q_n^l (query), k_n^l (key) and v_n^l (value) are computed for each s_n^l. Hence, q_n^l is taken dot product with each k_j^l $(1 \leq j \leq N)$ (with scaling the results by $\frac{1}{\sqrt{dim(q_n^l)}}$) and then softmax-normalised to yield the attention score $(\alpha_{n,N}^l)'$ as explained in Chapter 2.2.3. Those attention scores are, again, used as weights, to weighted-sum $\{v_j^l \mid 1 \leq j \leq n\}$ into $[(s_1^{l+1})', (s_2^{l+1})', \cdots, (s_N^{l+1})']$. The new sequence is then fed to a feedforward neural network which carries out identical operations on all the input positions and then generates the output sequence $[s_1^{l+1}, s_2^{l+1}, \cdots, s_N^{l+1}]$. By stacking several such self-attention blocks up and linking the components with residual connections, the transformer is able to learn at the top layer a semantic representation of the input sentence with self-attention.



(a) Overall Structure **(b)** Attention **(c)** Multi-Head Attention

Figure 2.8: Transformer. Source: Vaswani et al. (2017)

To reduce bias, the model also utilises what it calls multi-head attention, as sketched in Figure 2.8c, where each q^l_n, k^l_n, and v^l_n are linearly projected M times with different, learned transformations into $\{q^l_{n,1}, q^l_{n,2}, \cdots, q^l_{n,M}\}$, $\{k^l_{n,1}, k^l_{n,2}, \cdots, k^l_{n,M}\}$, and $\{v^l_{n,1}, v^l_{n,2}, \cdots, v^l_{n,M}\}$. The m^{th} head then is responsible for conducting the m^{th} attention scheme based on $\{q^l_{i,m} \mid 1 \leq i \leq N\}$, $\{k^l_{i,m} \mid 1 \leq i \leq N\}$, and $\{v^l_{i,m} \mid 1 \leq i \leq N\}$. Finally, $\{s^{l+1}_{n,1}, s^{l+1}_{n,2}, \cdots, s^{l+1}_{n,M}\}$ are concatenated and then projected (again, with a learned matrix) into s^{l+1}_n. The multi-head attention procedure thus diversifies attention.

Figure 2.8a illustrates the overall structure of the transformer, where on the left the encoder works exactly in the aforementioned manner while also adding a positional encoding to each input position — after all, attention is order-agnostic, therefore the positional encoding introduces necessary information about the token ordering into the process.

On the right is a decoder block, which is autoregressive. It also starts with positional encoding and then a self attention unit, but this time with masking to assign negative infinity to v's for all the input positions not filled by past outputs. This trick makes sure no attention is paid to the "future".

A cross-attention unit then follows in the decoder block, differing from self attention only in the source of the queries, keys, and values used. As the arrows in Figure 2.8a, the queries still come from the previous-layer input sequence, whereas the keys and values are from the encoder top-layer output, thus connecting the encoder and decoder and essentially building the output of the unit out of materials from the encoder.

Hence, a self-attention unit, a cross-attention unit, and a feedforward network constitute a decoder block with residual connections. By building up a number of such blocks preceding a (linear transformation → softmax) layer, a word is generated at the top layer. The word then fills the next position of the overall sequence, and the latter is used as input to the bottom decoder block for the next decoding step.

Since its advent, the transformer has enabled considerable efficiency thanks to its unique parallelism-friendly attention computation, and most important of all has surpassed RNN-based models in many MT scenarios. It is one of the most popular models among NMT researchers, and variants of the architecture are dominating leaderboards.

2.2.5 Deliberation

There is a weakness present in both RNN-based MT models and the transformer: the decoder is autoregressive and hence only utilises the output it has hitherto generated, which is not exactly how a human translator does their job: produce a translation draft, and then refine it before reaching the final result.

A deliberation network (Xia et al., 2017) simulates this refinement process and achieves state-of-the-art results. As shown in Figure 2.9, an RNN-based (the deliberation idea is model-independent, so it can be GRU- or LSTM-based too) deliberation network features a second pass decoder for refinement, in addition to an encoder and a first-pass decoder with identical functionality to the conventional encoder-decoder-with-attention structure introduced in Chapters 2.2.1 and 2.2.3.

Specifically, the encoder $\mathcal{E}$ finishes its encoding $\{h_1, h_2, \cdots, h_{T_x}\}$, before the first pass decoder $\mathcal{D}_1$ attends to those encodings while autoregressively generates a first pass translation $\{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{T_{\hat{y}}}\}$ and the corresonding before-softmax hidden states $\{\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_{T_{\hat{y}}}\}$. The second pass decoder $\mathcal{D}_2$ begins its task at this point, autoregressively decoding while attending separately to both $\{h_1, h_2, \cdots, h_{T_x}\}$ and $\{[\hat{s}_1; \hat{y}_1], [\hat{s}_2; \hat{y}_2], \cdots, [\hat{s}_{T_{\hat{y}}}; \hat{y}_{T_{\hat{y}}}]\}$, where the latter is the sequence of concatenations of hidden states and their token outputs. Two context vectors $ctx_e^{'}$ and ctx_e are produced as a result, and they are joint inputs with s_{t-1} (previous-time-step $\mathcal{D}_2$ hidden state) and y_{t-1} (previous-time-step $\mathcal{D}_s$ output) to $\mathcal{D}_2$ to yield s_t and then y_t.

A transformer-based deliberation architecture is proposed by Hassan et al. (2018), as Figure 2.10 depicts. It follows the same two-pass refinement process, with every second-pass decoder block attending to both the encoder output $\mathcal{H}$ and the first-pass before-softmax hidden states $\hat{\mathcal{S}}$. However, it differs from Xia et al. (2017) in that the actual first-pass translation $\hat{y}$.

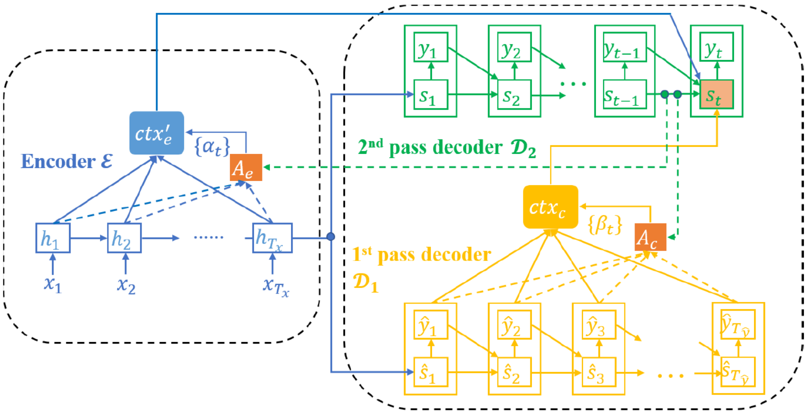


Figure 2.9: RNN-based Deliberation. Source: Xia et al. (2017)

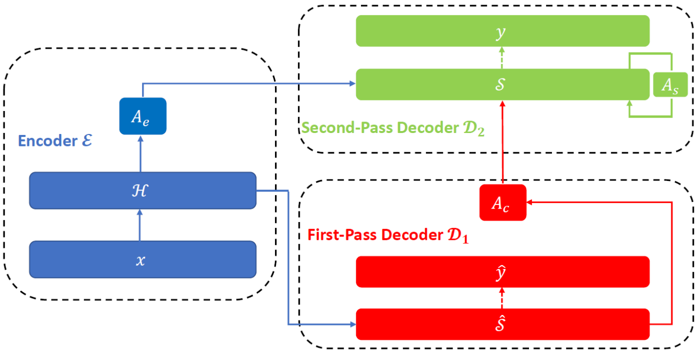


Figure 2.10: Transformer with Deliberation. Source: Hassan et al. (2018)

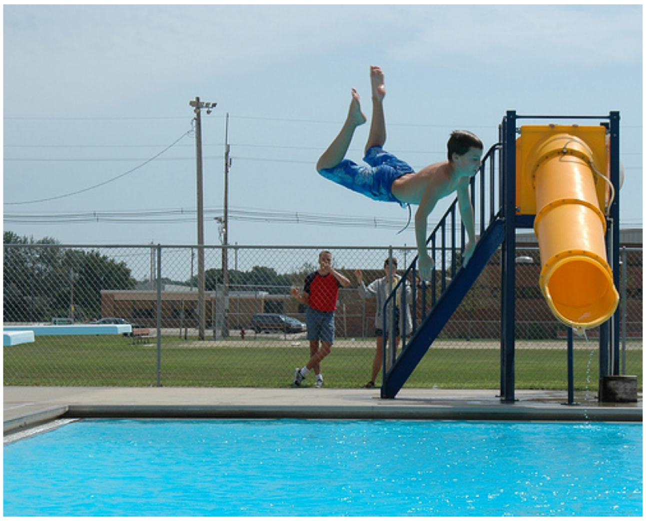
2.3 Neural Multimodal Machine Translation

Neural multimodal machine translation, as mentioned before, is based on NMT and primarily focused on ways of integrating visual clues into NMT models, and Chapter 2.3.1 covers the essentials of the topic. A broad categorisation of (N)MMT models according to those ways is: visual context as a single global vector, as spatial features, and as objects. Chapters 2.3.2, 2.3.3, and 2.3.4 are dedicated to the three categories respectively. A selection of models in each category are visited, the majority of which employs the attention mechanism and are GRU- or LSTM-based with optional bidirectionality and multilayering (see Chapters 2.2.3 and 2.2.2 for more details). These details will not be specified for the remainder of this chapter unless necessary.

2.3.1 Problem

Up until recently, multimodal machine translation has been in the form specified by the Share Task on Multimodal Machine Translation (Specia et al., 2016; Elliott et al., 2017a; Barrault et al., 2018a): given an image and its description in English, translate the description into a target language. The research work introduced in Chapters 2.3.2, 2.3.3 and 2.3.4 largely follows this setup.

The mostly commonly used dataset for MMT research, also the official dataset of the Shared Task, is Multi30K (Elliott et al., 2016), where an example is an (Image, English description, Target-language translation), as shown in Figure 2.11. The images and their English descrpitions come from Flickr30K (Young et al., 2014), while the target-language translation was obtained through crowdsourcing. The translations were only in German at first, but were then extended to include French and Czech. For the 2018 Shared Task, the dataset split was 29,000 examples for training, 1,014 for validation, and 1,071 for testing.



En: A boy dives into a pool near a water slide.
De: Ein Junge taucht in der Nähe einer Wasserrutsche in ein Schwimmbecken.
Fr: Un garçon plonge dans une piscine près d'un toboggan.
Cs: Chlapec skáče do bazénu poblíž skluzavky.

Figure 2.11: An example of the Multi30K dataset, with an image, its English description from Flickr30K, and the crowdsourced translations in German, French and Czech. Source: Barrault et al. (2018a)

Other datasets have also been used for related purposes. For example, MSCOCO (Lin et al., 2014) has been utilised for training in unconstrained settings (e.g. Helcl et al. (2018)), where a model has access to out-of-domain (i.e. non-Multi30k) data to train its specific components, such as object detection.

MMT metrics are in general directly from NMT. Of those, BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007) are the most reported in MMT research work, and the latter is favoured by the Shared Task. BLEU (BiLingual Evaluation Understudy) is a precision-based automatic metric and among the most established in NMT, evaluating the similarity between two sentences based on their constituent n-grams. Reported scores usually take into account unigram-, bigram-, trigram- and 4-gram-level BLEU, on the grounds that lower-order BLEU reveals translation quality on the lexical level whereas higher-order BLEU shows how well a model performs on a chunk/sentence basis. A brevity penalty is also given by BLEU to reduce partiality towards translations that omit parts of the source sentence but are n-gram-wise accurate. Since BLEU otherwise disregards recall, the brevity penalty mechanism alleviates this issue.

Unlike BLEU, METEOR (Metric for Evaluation of Translation with Explicit ORdering) assigns a significantly more important role to recall by rewarding long segments in the candidate and reference translations that are in alignment. METEOR can also be language-dependent by considering synonyms and stems with the help of lexical databases such as WordNet (Miller, 1995).

Finally, lexical translation accuracy (Lala and Specia, 2018) was also adopted as a metric in the 2018 Shared Task to assess the translation of ambiguous words, based on the observation that multimodality tends to be substantially helpful for disambiguation.

2.3.2 As Single Global Vector

The arguably most straightforward manner of incorporating the visual context is to present a global, vectorial summary of an image to the MMT model. This type of visual context is also readily available: the output of a fully-connected layer of the softmax layer of a convolutional neural network (CNN) (LeCun et al., 1989) operating on images usually suffices and is the typical choice. As a high-level summary of the picture, the vectorial summary contains little to no information about local details.

Elliott et al. (2015) proposed a classic approach, shown in Figure 2.12 where the fully-connected layer output is taken from a CNN and then initialises the encoder and optionally the decoder. The solid arrows in the figure indicate inputs that must be provided, whereas the dashed arrows represent optional ones. This setup enables a number of different models varying in their initialisation methods, including whether to use the vectorial summary as auxiliary initialisation input for the encoder as well as whether to initialise the decoder with the source encodings or the visual summary or both. A fully connected neural network is in place to process the textual and visual information when both are involved, so that the output of the network is used as the initialising vector.

Specifically, the features used in Elliott et al. (2015) come from the penultimate layer of pre-trained VGG-16 (Simonyan and Zisserman, 2014), a CNN model for image recognition. Without attention employed at all for the source sentence, the models were used for MMT and the configuration of the optimal model was found to be initialising the encoder with auxiliary visual input and initialising the decoder with only the encoding. It was also observed that the improvement brought by this model over the text-only baseline was most pronounced where the latter generated poor translation.

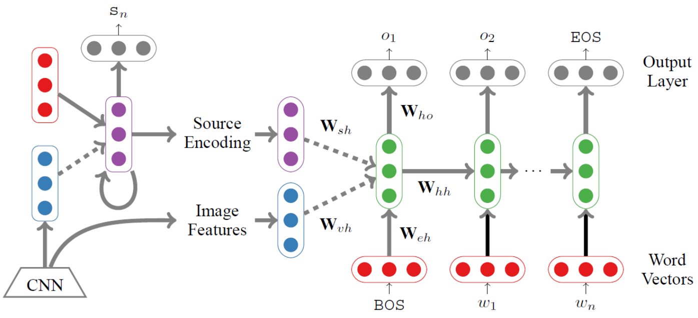


Figure 2.12: Encoder/Decoder Initialisation with Fully-Connected Layer Output. Source: Elliott et al. (2015)

Softmax features, on the other hand, are used in Madhyastha et al. (2017), as depicted Figure 2.13. The intuition behind this idea is that softmax probabilities represent high-level semantics and can therefore better aid translation. The 1000-dimensional softmax features in this case are extracted from the 152-layer ResNet (He et al., 2016), also an image recognition network. While the features are also used for encoder/decoder initialisation similar to Elliott et al. (2015), Madhyastha et al. (2017) additionally weighted-sum the word embeddings of the 1000 image categories with their softmax probabilities (e.g. $0.90 \times dog + 0.05 \times cat + 0.03 \times fox + \cdots + 0.00 \times apple$), and then add the result to each word in the source sentence. As for the statistics, the conclusion was that the model with a multimodally initialised decoder had the best performance, contradicting the observation by Elliott et al. (2015).

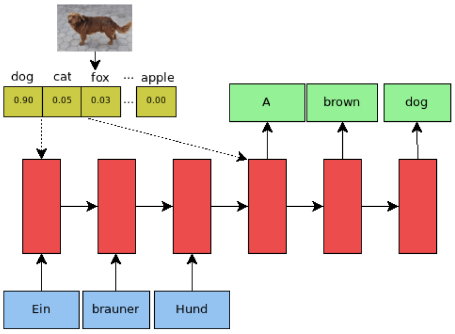
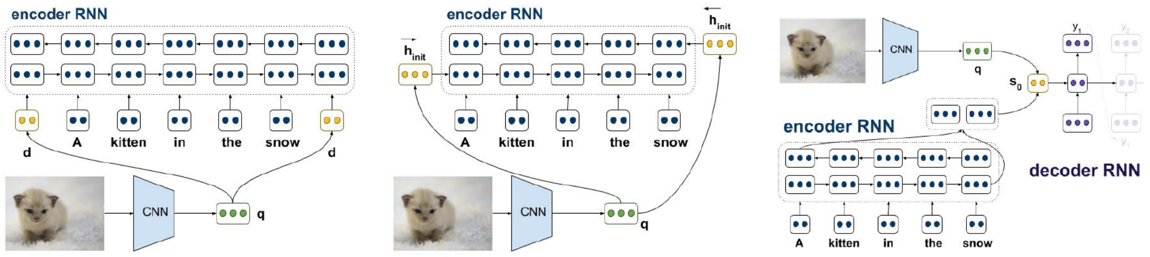


Figure 2.13: Encoder/Decoder Initialisation with Softmax Layer Output. Source: Madhyastha et al. (2017)

Word embedding modulation is another option for utilising the vectorial summary, suggest Caglayan et al. (2017). In this scenario, an embedding gate g is obtained as a result of $tanh(W_{img} \cdot V)$, where V is 2048-dimensional taken from the pool5 layer of ResNet-50 (He et al., 2016) and W_{img} is a learned projection matrix that maps from the visual space to the embedding space. The gate has the same dimensionality as a word embedding does and is hence element-wise applied to every word for modulation. Caglayan et al. (2017) tried combinations of word modulation on the source/target side and multimodal encoder/decoder initialisation, and found that (target-word modulation + unimodal initialisation) achieved the highest METEOR scores but was considered worse than the baseline according to BLEU, revealing a metric discrepancy.

Also based on mapping the visual summary to the embedding space, Calixto et al. (2017b) conduct the transformation and then insert the result as the first and/or last word(s) of the source sentence (Figure 2.14(a)), as well as multimodal encoder (Figure 2.14(b)) and decoder (Figure 2.14(c)) initialisation. The visual features are from the second fully-connected layer of VGG-19 (Simonyan and Zisserman, 2014) and mapped to the embedding space via a two-layer feedforward network. The first approach proved to give consistently inferior performance than the initialisation-based models, and, again, multimodal initialisation for only the encoder or decoder was found to be better than for both.



(a) An encoder bidirectional RNN that uses image features as words in the source sequence.

(b) Using an image to initialise the encoder hidden states.

(c) Image as additional data to initialise the decoder hidden state s_0.

Figure 2.14: Visual Summary as Additional Word(s) / Initialiser. Source: Calixto et al. (2017b)

Imagination, proposed by Elliott and Kádár (2017), is drastically different from the methods mentioned so far, in that it is a multitask learning process with no visual input during decoding. A unique feature of the model is a shared encoder $\mathcal{E}$ whose output is fed to a translation decoder $\mathcal{D}_{\mathcal{T}}$ as well as a visual decoder $\mathcal{D}_{\mathcal{V}}$, as shown in Figure 2.15. During training, $\mathcal{E}$ accepts a source sentence as input, the entirety of its hidden states $\mathcal{H}$ utilised via attention by $\mathcal{D}_{\mathcal{T}}$ to produce a translation while $\bar{\mathcal{H}}$, the average, is given to $\mathcal{D}_{\mathcal{V}}$, a feedforward network that generates a vector in the visual feature space to approximate the true visual features from a CNN (hence the name "imagination"). Two training goals are optimised towards: (a) minimise the translation loss (negative log likelihood of the translation decoder producing the correct translation) (b) minimise a margin-based difference between the true visual features and the ones "imagined" by $\mathcal{D}_{\mathcal{V}}$.

During training, the two goals interleave in cycles, where in each cycle the model is first trained towards the translation loss until convergence (chief objective) and then optimised based on the imagination loss (secondary objective). For decoding, as mentioned before, $\mathcal{D}_{\mathcal{V}}$ is not used at all, with only $\mathcal{E}$ feeding $\mathcal{D}_{\mathcal{T}}$ to generate translations. What this process effectively achieves is that the encoder learns to represent a source sentence in a visually-grounded fashion, which allows the model to exhibit competitive translation performance compared to other MMT models despite its having no access to the visual context. This merit is especially relevant when out-of-domain data is available for pre-training $\mathcal{E}$ and $\mathcal{D}_{\mathcal{T}}$.

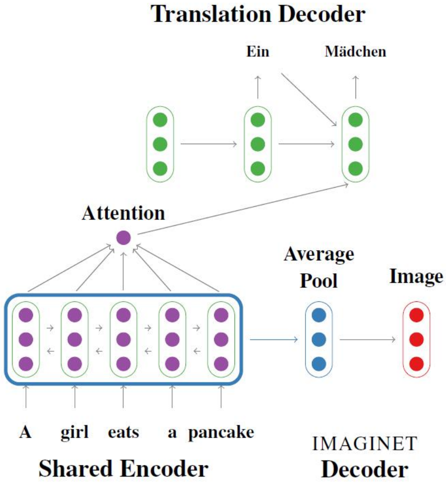


Figure 2.15: Imagination Model. Source: Elliott and Kádár (2017)

A transformer-based imagination model is put forward by Helcl et al. (2018), which also witnessed substantial improvement of MMT quality with pre-trained $\mathcal{E}$ and $\mathcal{D}_{\mathcal{T}}$.

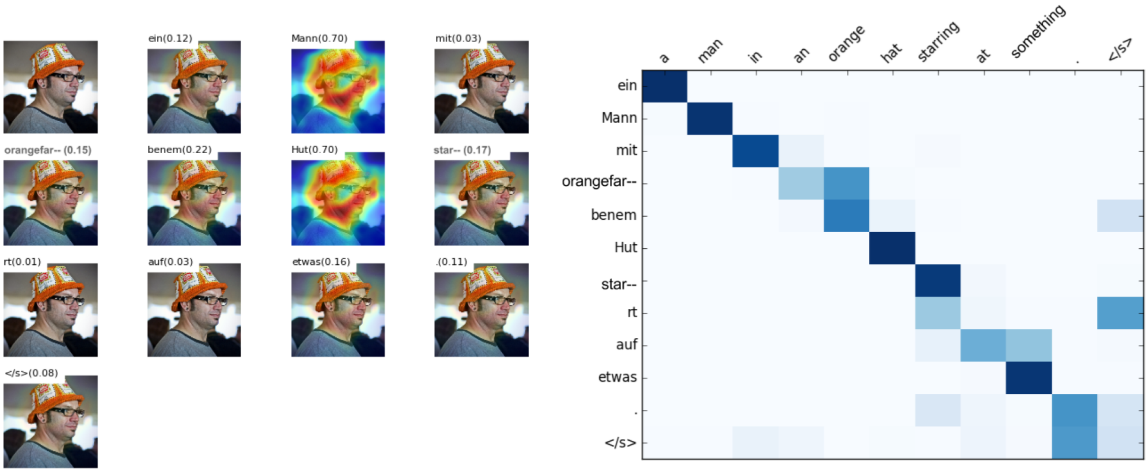
Calixto et al. (2019), on the other hand, propose a latent-variable alternative that also achieves visually grounded translation model during training time and does not need images at inference time. In this approach, an image and a translation are seen as independently generated given a "common stochastic embedding" z, such that z can be drawn from a latent Gaussian distribution given a source sentence s, and an image vector v is in turn drawn from a Gaussian observational distribution given z. As for the translation t, each target-language word is drawn from a categorical distribution formed by the softmax probabilities produced by an attention-based decoder that takes z as an additional input at every time step. All the distributions in this variational model are parameterised by feedforward neural networks, except the categorical distribution which is by a seq2seq model. This variational approach is shown to outperform Imagination on Multi30K.

2.3.3 As Spatial Features

As applying attention over the words in an input sentence has enabled significant improvement in MMT, so has visual attention over the regions in an image. In this scenario, the visual features are usually the features maps measuring $C \times N \times N$ extracted at a convolutional layer of a CNN, which can be seen as the image being divided as a grid of $N \times N$ cells, each a C-dimensional summary of the corresponding region in the image. The visual attention is usually distributed across the $N \times N$ regions, at which the earliest attempt was Xu et al. (2015), where the image captioning model learns to focus on (a) particular region(s) of an image when it is generating the next word (e.g. looking at the bird in the image helps it come up with the word "bird"). Spatial-features-based MMT models mostly operate on this basis.

An intuitive way of applying the visual attention in MMT is to execute the context-vector procedure introduced in Chapter 2.2.3: a feedforward network computes a attention score for each region w.r.t. the decoder hidden state, and the weighted sum of those region representations ($N \times N$ in total) is calculated as the visual context vector. As for how to integrate the visual and textual context vectors, Caglayan et al. (2017) choose plain concatenation and then perform translation as before based on the fused context vector.

Calixto et al. (2017a) propose a model in a similar vein, except that it introduces a time-dependent gate to control the influence of the visual context vector. Specifically, the visual and textual context vectors are calculated as before, but then a feedforward network is in place to produce a gate, only taking the hidden state from the previous time step. The visual context vector is then multiplied with the gate to yield its final value. The idea behind the trick is that words with little presence in the visual domain, such as "a" and "i" as opposed to "orange", do not need visual clues to confuse the decoder. With this method, the model was found to enable improved performance especially when measured by recall-oriented metrics. As shown in the example in Figure 2.16, the ungated textual attention is properly distributed according to the German-English lexical relations, whereas the only two words that utillise gated visual attention for MMT are Mann (man) and Hut (hat), both with strong presence in the image. This clearly shows the gating mechanism is successful in incorporating visual attention only when necessary.



(a) Image–target word alignments.

(b) Source–target word alignments.

Figure 2.16: Gated Visual Attention & Ungated Textual Attention. Source: Calixto et al. (2017a)

Delbrouck and Dupont (2017) explore beyond the weighted-sum context vector — they call it "soft attention" — and focus on two more attention schemes: hard stochastic attention and local attention. For the former, a multinoulli distribution is established according to the attention weights calculated the same way as before, and, instead of weighted-summing, the context vector is simply one region in its original C-dimensional vectorial form sampled from the distribution out of the total $N \times N$ at each decoding step. Local attention, on the other hand, is in essence choosing a small patch around the region sampled by hard attention. The time-dependent gating scalar introduced by Calixto et al. (2017a) is also employed in all the models in this work. Despite the hard-attention model risking ignoring inter-region interactions, the results show that it consistently outperformed the others. It was also found that the gating scalar at many decoding steps was small even for the hard-attention model, which means that the visual context was ignored almost entirely for translating those words.

Two other options are investigated by Libovickỳ and Helcl (2017): flat attention combination and hierarchical attention combination. To achieve the former, all the visual representations H_V ($N \times N$ regions in total, C-dimensional each) and source sentence encoder hidden states H_E (L in total, S-dimensional each, where L is the sentence length and S the hidden state dimensionality) are mapped to a common semantic space S, with projection matrices W_V for H_V and W_E for H_E. The weighted-sum context-vector procedure is then applied in S to all the mapped vectors. For hierarchical attention combination, the visual and textual context vectors are computed the conventional way, and then two time-step-dependent scalars are used to weighted-sum the context vectors to yield the ultimate context vector. As for the performance, hierarchical attention combination was found to beat flat attention combination both in terms of automatic-metric results and convergence speed.

Without fusing the visual and textual context vectors directly, Helcl et al. (2018) utilise the hierarchical nature of the transformer decoder to achieve multimodal context integration. The main feature of this model is the multimodal decoder shown in Figure 2.17, where a visual cross-attention layer is inserted in between the textual cross-attention layer and feedforward layer with residual connections in each decoder block. The queries (Q) of this layer come from the output of the textual cross-attention layer below whereas the keys (K) and values (V) are generated from projected image region representations. In doing so, the model grounds the decoder hidden state sequence with both the source sentence (textually) and the region features (visually) before feeding it to the feed-forward layer. With competitive scores, the model also exhibited degraded performance when given randomly selected "fake" images, thus showing that it had truly learned to exploit visual information to help with translation.

2.3.4 As Objects

It can be argued that objects are a better medium of conveying visual information for MMT, especially for a dataset like Multi30K where an image description is often about interactions between objects, such as "a man in a vest is sitting in a chair and holding magazines". As a result, object-based MMT models have gained popularity

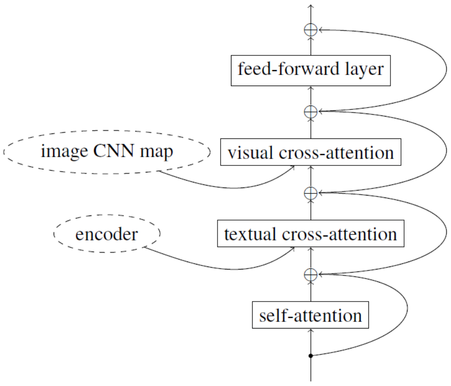


Figure 2.17: Multimodal Transformer. Source: Helcl et al. (2018)

in recent years, many of them utilising bounding-box or object-category features.

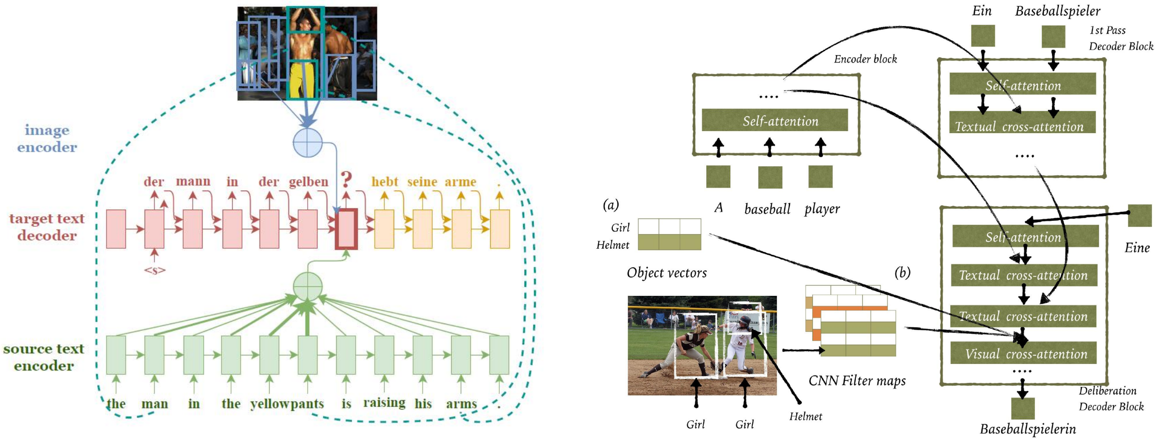
A pioneer model is proposed by Huang et al. (2016), where the region proposal network (RPN) in the region-based CNN (RCNN) (Ren et al., 2015) is used to detect objects and extract bounding boxes around them. The largest four bounding boxes are selected and then provided to VGG-19 (Simonyan and Zisserman, 2014) together with the original overall image features in their entirety. The outputs of the last fully-connected layer for those five bounding boxes are taken and mapped to the word embedding space and, as pseudo words, added at the beginning of the source sentence. The enhanced source sentence is then decoded with the usual text-based attention mechanism. The observation is that the model brought modest improvement compared to the text-only baseline.

Grönroos et al. (2018) on the other hand use Mask R-CNN (He et al., 2017) to generate a segmentation mask for each of the 80 MSCOCO (Lin et al., 2014) object categories, based on which an 80-D vector is produced as the visual features. A feedforward network then takes the object-level features along with the hidden state at a position of the encoder output or the decoder input to the softmax layer, so as to compute a gate to be element-wise applied to the hidden state. Thus is the modulation: a time-dependent object-based gating scheme executed for the encoder output or the decoder pre-softmax distribution. With the transformer as the underlying structure, Grönroos et al. (2018) found the model to lead to relatively small improvement, as substituting the averaged visual features for the correct ones did not give rise to considerable difference score-wise.

To associate the objects in an image to the words in the description more closely, Specia et al. (2019) devise an unsupervised object-to-word alignment strategy which matches a word with the object, among all the detected, whose label has the largest cosine similarity with the word in the embedding space.

Then, Explicit Referential Grounding (ERG) (Figure 2.18a) concatenates each word in the source sentence with the label of its paired object identified using the aforementioned strategy, or with the CCA (Hotelling, 1992) image feature projection of the object based on its label.

For Implicit Referential Grounding (IRG), each source word and each visual feature go through the trainable co-attention procedure (Lu et al., 2016) to yield new correlated representations of themselves modulated by the other, achieving object-level grounding on the source side. Hierarchical attention (Libovickỳ and Helcl, 2017) then operates on the new textual and visual representations for translation. The alignment strategy is optional for IRG in that it defines an auxiliary grounding loss inspired by Rohrbach et al. (2016) penalising cases where the highest co-attention weights are allocated to objects that are not the ground truth (i.e. not the ones found by the strategy).



(a) Explicit Referential Grounding. Source: Specia et al. (2019)

(b) MMT Deliberation. Source: Ive et al. (2019)

Figure 2.18: Two Models Based on Attention over Objects

Ive et al. (2019) propose a deliberation-transformer-based MMT model to refine the preliminary translation from a text-only decoder with a multimodal second-pass decoder. The intuition behind the architecture is the general finding within the MMT community that visual context is helpful for translation only in specific types of scenarios such as in the presence of ambiguity and from-genderless-language-to-gendered translation, which makes it sensible to have a quality unimodal translation first and then improve it based on visual information.

As shown in Figure 2.18b, each second pass decoder has a conventional self-attention layer and a textual cross attention layer w.r.t. the source sentence, then another textual cross attention layer follows, attending to the first pass translation, before sending the output to the subsequent visual cross attention layer that attends to the image features.

The visual features used are an N-hot object label embedding matrix per example. Specifically, an object detector (Kuznetsova et al., 2018) decides N out of the total 545 categories of objects are present in the image, and the model therefore only attends to the 50-D GLoVe embeddings Pennington et al. (2014) of the N categories. In their experiments, Ive et al. (2019) show that visual information for deliberation was particularly helpful when the source sentence was noisy or needed considerable restructuring to be translated into the target language.

Chapter 3

Design

In this chapter, the details of our multimodal translation models and multimodal features are explained. As mentioned in previous chapters, we focus on two scenarios: translation with and without English subtitles. For the former, we directly apply multimodal transformers that exploit three types of visual features (`VS`, `CLO`, `ACE`) in two ways (`Enc-AVC`, `Dec-AVF`) to aid translation. For the latter scenario, an off-the-shelf speech recognition system is employed to yield transcripts, and then multimodal translation of the transcripts is carried out by multimodal transformers and deliberation networks. We begin by introducing the speech recognition system, then move on to visual feature extraction, and finally elaborate on our multimodal transformers and deliberation networks.

3.1 Automatic Speech Recognition

We use the unimodal baseline ASR model provided by Caglayan et al. (2019b) to obtain the transcripts for the training, validation and test set audios. The ASR model consists of a 6-layer bidirectional LSTM-based encoder and a 2-layer GRU-based decoder.

At the encoder side, $tanh$ projection is applied in between each encoder layer, and the 2^{nd} and 3^{rd} encoder layers execute temporal subsampling (Chan et al., 2016) which shrinks the sequence of speech features to 1/4 of its original length by jumping two input positions at each time step. Each layer comprises of 320-dimensional hidden states, hence the final encoder outputs $\mathcal{H}^{E}_{ASR}$, after the temporal subsampling, are a sequence of 320-D encodings 1/4 of the original length.

The first (bottom) GRU layer of the decoder is first initialised with the mean of the all the encoder outputs and then autoregressively generates its hidden state $\mathcal{H}^{D_1}_{ASR}(t)$ at decoding step t. A feedforward network is employed, taking $\mathcal{H}^{D_1}_{ASR}(t)$ and $\mathcal{H}^{E}_{ASR}$ as input, to produce a context vector z^{t}_{ASR} which, along with $\mathcal{H}^{D_1}_{ASR}(t)$, is fed to the second (top) GRU layer of the decoder. A $tanh$-based fully-connected layer then processes the top GRU layer hidden states, followed by a linear transformation and finally a softmax layer that deciphers the result into a word based on a vocabulary.

3.2 Visual Features

It is notable that videos differ from images that the former capture actions much better, which means a video-based action recognition network can contain rich semantic information about the video. Following this direction, we obtain three types of video visual information as feature maps from two such networks.

3.2.1 VideoSum (VS)

We use off-the-shelf VS features provided by the How2 Challenge (Sanabria et al., 2018) for the videos on which we operate. In this approach, a video is segmented into smaller parts of 16 frames each, and the segments are fed to ResNeXt-101 (Xie et al., 2017), a CNN with 3D convolutional kernels trained for recognition of 400 classes of actions (Hara et al., 2018). The 2048-D feature maps at the fully-connected layer of the network when given the video segments are averaged as the final VS features for the video. Therefore, a 2048-D vectorial feature can be seen as a high-level summary of a video, similar to the single-global-vector approach described in Chapter 2.3.2.

3.2.2 Convolutional Layer Output (CLO)

In order to have richer visual context, we apply 3D ResNet-50 (Monfort et al., 2019) network to each video and obtain the output of the last convolutional layer. Specifically, 16 equi-distant frames are sampled for a video, and they are then used as input to the network. The network is based on the ResNet-50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009), inflating the originally 2D network into 3D and fine-tuned on the Moments in Time action video dataset (Monfort et al., 2019). 3D ResNet-50 hence takes in a video and classifies it into one of 339 categories. The CLO feature, taken at the $conv4$ layer of the network, has a $7 \times 7 \times 2048$ dimensionality, which can be interpreted as dividing a video spatially into a 7×7 grid of 49 regions where each region is temporally summarised as a 2048-D vector, suitable for the spatial-feature MMT strategy introduced in Chapter 2.3.3.

3.2.3 Action Category Embedding (ACE)

Higher-level semantic information can be more helpful than convolutional features. With that in mind, we apply the same 3D ResNet-50 network to a video as we do for CLO features, but this time the focus is on the softmax layer output. The original feature is a 339-D probability vector, and we process it in two ways:

(i) We multiply the 300-D CBOW word2vec embedding (Mikolov et al., 2013) of each category label with its softmax posterior prediction, thus obtaining a 339×300 matrix for each video where each row is a probability-scaled word embedding vector. We call it the Probability-Scaled Action Category Embedding (PSACE) features.

(ii) We keep the word embeddings for the 10 categories of the highest softmax probabilities while assigning all-zero 300-D vectors to the other categories. We call it the Ten-Hot Action Category Embedding (THACE) features.

3.3 Transformers

The first type of our machine translation models is transformers. The vanilla transformer decribed in Chapter 2.2.4 is used as a baseline, and we design two variants: with additive visual conditioning and with attention to visual features, which rely on an encoder with additive visual conditioning and a decoder with attention to visual features respectively.

3.3.1 Encoder with Additive Visual Conditioning (Enc-AVC)

In this approach, inspired by Ive et al. (2019), we add a projection of the visual features to each output of the vanilla transformer encoder, the latter introduced in Chapter 2.2.4. This projection is strictly linear from the 2048-D VideoSum features to the 1024-D space in which the self attention hidden states reside, and the projection matrix is learned jointly with the transformer/deliberation model.

3.3.2 Decoder with Attention to Visual Features (Dec-AVF)

In order to accommodate attention to visual features at the decoder side and inspired by Helcl et al. (2018), we insert one layer of visual cross attention at each vanilla transformer decoder block, after the textual cross attention layer and before the fully-connected layer. The keys and values are therefore from the visual features whereas the queries are from the hidden states of the textual cross attention layer.

The visual features in this decoder come from one of the three sources: VideoSum, Convolutional Layer Output, and Action Category Embedding.

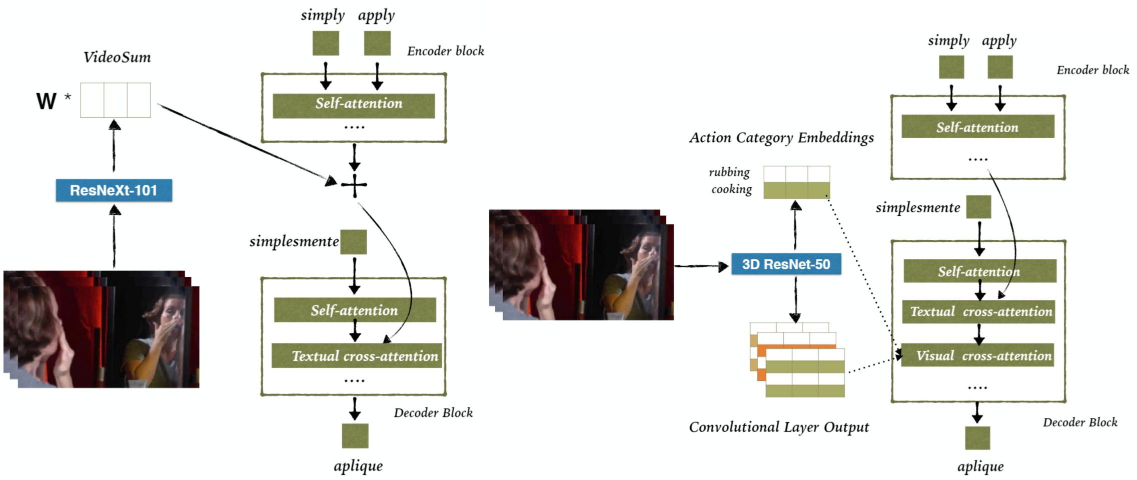
(i) For VideoSum, which is originally 1×2048 dimensional, we reshape it into 32×64-D in row-major order to explore whether segments of fully-connected layer output of a convolutional neural network can offer more insight than using a global vector as a whole. Naturally, the attention is over the 32 rows of the reshaped features.

(ii) For Convolutional Layer Output, the $7 \times 7 \times 2048$ dimensional tensor is reshaped into 49×2048-D, so that the decoder attends to the 49 regions of the video frames at the visual cross attention layer.

(iii) For Action Category Embedding, the features themselves attention-ready, i.e. the attention of the decoder is distributed across the 339 action categories, whether this features are probability-scaled or ten-hot. For the later, it is expected that the categories with all-zero rows will enjoy no attention at all, hence the decoder only attends to the action categories detected.

3.3.3 Transformer with Additive Visual Conditioning (`Trans-AVC`)

This variant features an encoder with additive visual conditioning detailed in Chapter 3.3.1 and a vanilla transformer decoder, therefore utilising visual information only at the encoder side. Figure 3.1a shows its structure.



(a) `Trans-AVC` Structure **(b)** `Trans-AVF` Structure

Figure 3.1: Two types of multimodal transformer. Note that VideoSum features from ResNeXt-101 can also be used for `AVF`, therefore the sketches are just for illustration purposes.

3.3.4 Transformer with Attention to Visual Features (`Trans-AVF`)

In contrast to TAVC, we configure this transformer model with a vanilla transformer encoder and a decoder with attention to visual features as detailed in Chapter 3.3.2. Obviously, visual cues this time are only for the decoder to exploit. Its structure is given in Figure 3.1b.

3.4 Deliberation Networks

As mentioned in Chapter 2.2.5, a deliberation network is aimed at refining the output of a first pass decoder. In this project, the deliberation is based on the transformer structure, as in Xia et al. (2017) and Ive et al. (2019).

For deliberation itself, there are two ways of integrating the textual attentions to the encoder output and to the first pass output: additive and cascade. For handling visual features, we employ additive visual conditioning or visual attention, as introduced in Chapters 3.3.3 and 3.3.4 respectively.

3.4.1 Additive or Cascade Deliberation at Second Pass

In an additive-deliberation second-pass decoder block, the first layer is still self-attention, whereas the second layer is the addition of two separate attention sub-layers. Specifically, the first sub-layer attends to the encoder output in the same way the vanilla transformer decoder does, while the attention of the second sub-layer is distributed across concatenated first pass outputs and hidden states. The input to both sub-layers is the output of the self-attention layer, and the outputs of the sub-layers are summed as the final output of the second layer and then (after a residual connection from the second layer) fed to the visual attention layer if the decoder is multimodal or to the fully connected layer otherwise.

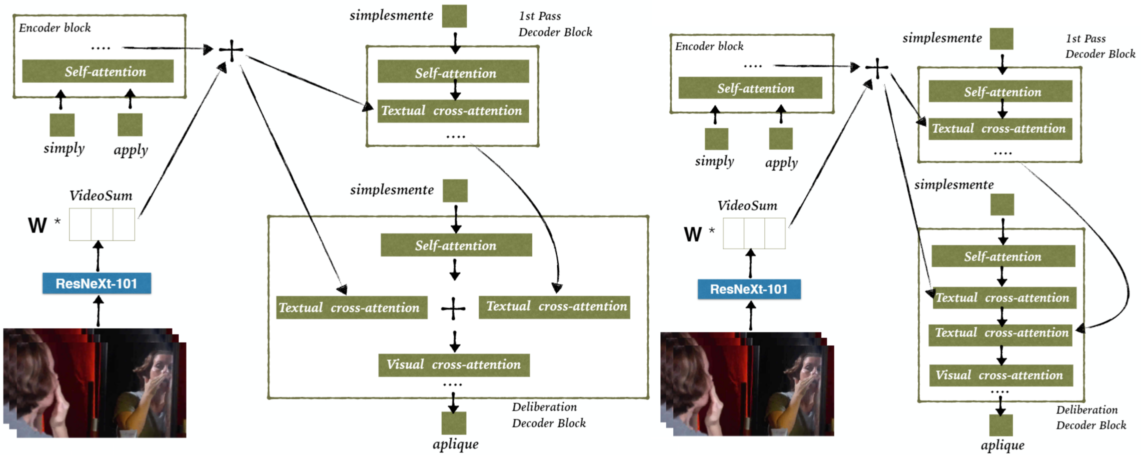
For cascade attention, the only difference from additive deliberation is that at the second layer, instead of having two sub-layers and summing their results, we separate them as two actual layers. In other words, the second layer in this model is textual cross attention to the encoder output while the third layer is textual cross attention to [first pass output; hidden state] concatenations.

3.4.2 Deliberation with Additive Visual Conditioning (`Delib-AVC`)

Similar to in Transformer with Additive Visual Conditioning, we add a projection of the visual features to the deliberation encoder (i.e. vanilla transformer encoder), and use the vanilla transformer decoder as the first pass decoder and either additive (Figure 3.2a) or cascade deliberation as the second decoder (Figure 3.2b).

3.4.3 Deliberation with Attention to Visual Features (`Delib-AVF`)

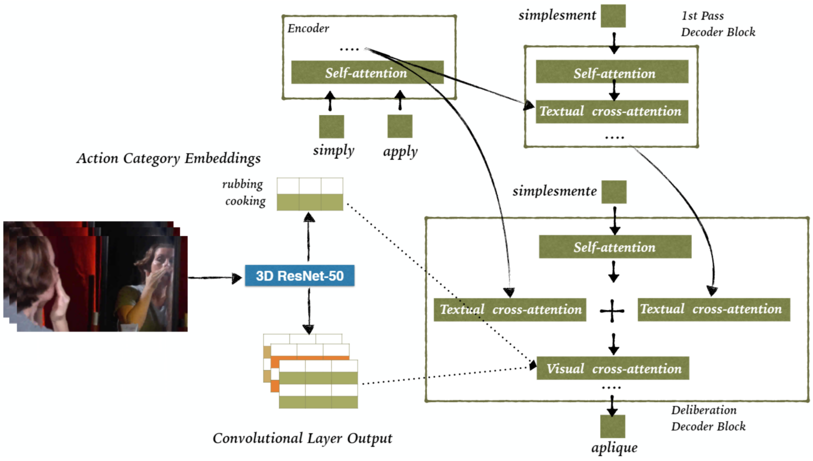
In a similar vein as `Trans-AVF`, the encoder in this setting is simply a vanilla transformer encoder with self attention and the first pass decoder also just involves textual cross attention to the encoder outputs, but this time the second pass decoder is responsible for attending to the first pass output as well as the visual features. For both additive (Figure 3.3a) and cascade (Figure 3.3b) deliberation, a visual attention layer (`Dec-AVF`, see Chapter 3.3.2) is inserted immediately before the fully-connected layer, so that now the penultimate layer of a decoder block attends to visual information.



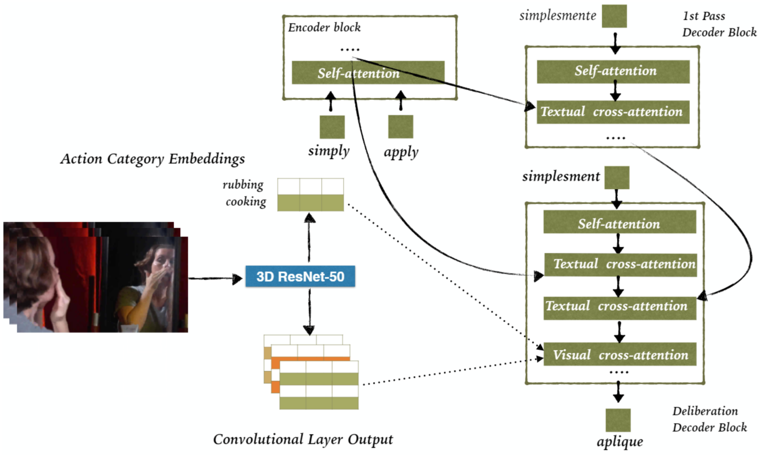
(a) `Delib-AVC-A` Structure

(b) `Delib-AVC-C` Structure

Figure 3.2: Two types of `Delib-AVC`: one (left) with additive deliberation, the other (right) with cascade deliberation



(a) `Delib-AVF-A` Structure



(b) `Delib-AVF-C` Structure

Figure 3.3: Two types of `Delib-AVF`: one (top) with additive deliberation, the other (bottom) with cascade deliberation

Chapter 4

Experiments

The focus of this project is to study the contribution of visual information to the translation of corrupted or incomplete source texts. Therefore, we conduct a **two-phased** experiment, with differently compromised source texts: in **Phase 1**, we train multimodal and text-only transformers to translate English subtitles with different degrees of verb masking; in **Phase 2**, multimodal transformers and deliberation networks are trained to translate transcripts of How2 videos into Portuguese. For **Phase 1**, verb masking achieves artificially corrupted source sentence, whereas transcribing based on speech recognition in **Phase 2** necessarily leads to information at the source side being left out or modified.

We begin this chapter by introducing the dataset (4.1), different types of sources (4.2) as well as some settings shared by both phases (4.3, 4.4), and then elaborate on the details of each phase in Chapters 4.5 and 4.6 respectively.

4.1 Dataset

The dataset for our experiments is How2 [1], a large-scale dataset for multimodal language understanding. It consists of English-language instructional videos segmented into smaller clips, and we keep its default splits: 184,949 video clips for training, 2,022 for validation, and 2,305 for testing. The clips are around 300 hours in duration in total, and each on average lasts around 5.8 seconds and has 20 words in its paired English subtitles. Each video segment also has its crowd-sourced Portuguese subtitles.

[1] `https://github.com/srvk/how2-dataset`

4.2 Source Domains

Our baseline model translates the original English subtitles as source text into Portuguese without any multimodal information. We explore several variants of the source for our experiments, as detailed below.

4.2.1 Verb Masking for Source Subtitles

Since motion is an important element of videos, it can be argued that verbs in subtitles are well represented in videos, and this is also pointed out by the creators of the How2 dataset (Sanabria et al., 2018) and hence their officially provided VideoSum features (extracted from an action recognition network, as described in Chapter 3.2.1). Therefore, we decide to explore whether visual information can be particularly helpful for translation in cases where some verbs are missing from the source text. We investigate three scenarios: Original Source (`ORG`), Mask Action Verbs (`ACT`) and Mask All Verbs (`ALL`).

Mask Action Verbs (`ACT`): each verb in the subtitles that is associated with one of the 339 action category labels defined in the Moments in Time dataset (Monfort et al., 2019) is replaced by a placeholder. Our statistics show that 2.75%, 2.83%, and 2.84% of the words (tokens) in the training, validation and test set source texts, respectively, are replaced in this setting.

Mask All Verbs (`ALL`): each verb in the subtitles is replaced by a placeholder. Our statistics show that 20.6%, 21.0%, and 20.4% of the words (tokens) in the training, validation and test set source texts, respectively, are replaced in this setting.

We first POS-tag the subtitles to mark the verbs (so that `ALL` masking can be achieved by simply replacing those masked) and then carry out lemmatisation, both with spaCy 2.0 [2]. The processed verb tokens are matched against the (also lemmatised) action category labels from Monfort et al. (2019), and the subtitle tokens associated with the labels are hence masked for `ACT` masking [3]. The Portuguese translations, on the other hand, remain unchanged throughout the experiments.

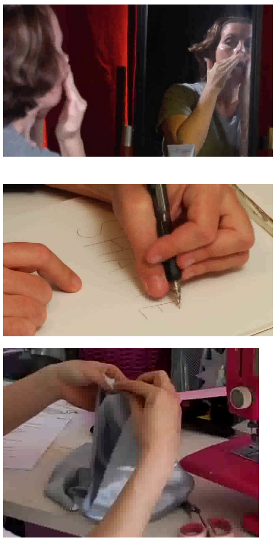
See Figure 4.1 for How2 examples with `ACT` and `ALL` masking as well as the original sentences.

4.2.2 Transcripts as Source

With the ASR model detailed in Chapter 3.1, we obtain lowercased punctuation-free transcripts. We do not apply any masking to them, since they are considered to inevitably contain random noise introduced during transcribing time.

[2] `http://spacy.io/` model `en_core_web_lg`

[3] We retain only the verb component for specialised actions such as *playing+music* and *adult+male+singing*



■ simply apply the cleanser or cream to your hands and apply it to the face and begin rubbing.
⧫ simply apply the cleanser or cream to your hands and apply it to the face and begin V .
▲ simply V the cleanser or cream to your hands and V it to the face and V V .

■ you can draw it really lightly , go back and erase it later .
⧫ you can V it really lightly , go back and erase it later .
▲ you V V it really lightly , V back and V it later .

■ what we are going to be doing is folding the top over and making a little casing the ribbon iwill slip through .
⧫ what we are going to be doing is V the top over and making a little casing the ribbon will V through .
▲ what we V V to V V V V the top over and V a little V the ribbon V V through .

Figure 4.1: Three verb masking examples from the How2 training dataset. In each one, the first line (■) shows the original sentence, the second (⧫) shows the sentence with action verbs replaced by V , and the third (▲) shows the sentence with all verbs replaced by V .

4.3 From Words to Vectors

It is worth noting that we do not employ any off-the-shelf word embeddings to encode our source and target sentence. Instead, in each source/target setting, an n-dimensional dictionary is established including all the unique words that occur in the corpus, then ⟨PAD⟩, ⟨EOS⟩ and ⟨UNK⟩ are inserted at the beginning to represent padding, end-of-sentence and unknown words, respectively. After the source and target dictionaries are prepared in this manner, every word is hence mapped to a one-hot vector and then projected to the hidden state space, with the corresponding embedding matrix jointly learned with the translation model.

4.4 Shared Training Hyperparameters

Our transformer and deliberation models are based upon the **transformer architecture** (Vaswani et al., 2017) implemented in the `tensor2tensor` [4] (Vaswani et al., 2018) library (1.3.0 RC1) as well as the vanilla transformer-based deliberation models[5] (Xia et al., 2017) and their multimodal variants[6] (Ive et al., 2019).

Like Ive et al. (2019), we train our transformer and deliberation models largely with `transformer_big` hyperparameters : 16 attention heads, 1024-D hidden states and a

[4] https://github.com/tensorflow/tensor2tensor
[5] https://github.com/ustctf/delibnet
[6] https://github.com/ImperialNLP/MMT-Delib

0.01 dropout rate during training time for layer pre- and post-processing. We apply size-10 beam search and an alpha of 0.1 in inference time, and record a checkpoint every 1,800 seconds for every model during training time.

As is pointed out by Popel and Bojar (2018), `tensor2tensor` defines only training steps (the number of (mini-)batches processed), as opposed to epochs (the number of times the whole training set is gone through). Also, the $batch\_size$ parameter in `tensor2tensor` is the number of tokens processed in one batch on one GPU, hence the effective batch size is $batch\_size \times \#GPUs$. Therefore, following the paradigm in Popel and Bojar (2018), we approximate the number of training epochs with the formula below for our training purposes:

$$epochs = \frac{steps \times batch\_size \times \#GPUs}{max(\#tokens\ in\ source\ corpus, \#tokens\ in\ target\ corpus)}$$

4.5 Phase 1: MMT with Verb Masking

4.5.1 Preprocessing

Phase 1 is a machine translation task, therefore we preprocess the English subtitles and Portuguese translations with the following pipeline: tokenisation → removing non-printing characters → replacing unicode punctuation marks with ASCII approximations → lowercasing → byte pair encoding (BPE) [7].

We use moses [8] for all but the final preprocessing steps. Specifically, tokenizer.perl, remove-non-printing-char.perl, replace-unicode-punctuation.perl and lowercase.perl are used for the first four steps. For BPE, we break down the tokenised English and Protuguese training texts (184,949 sentences in each masking setting) into sub-word units separately with 20,000 merge operations, leading to 18,963, 18,920, and 18,405 unique sub-tokens for `ORG`, `ACT`, and `ALL` respectively. Those unique sub-tokens are then kept in their entirety and collected as one distinct dictionary for each setting. Similarly, we obtain a dictionary of 19,499 sub-tokens for Portuguese translations. With the BPE codes for 20,000 merge operations, we then conduct BPE on the validation and test sets, hence finishing our preprocesing. Note: we do not share vocabularies between the source and target domains.

4.5.2 Models

For Phase 1, we experiment with five model variants:

(i) text-only transformer (`Trans-TO`)

[7] `https://github.com/rsennrich/subword-nmt` (Sennrich et al., 2015)

[8] `http://www.statmt.org/moses/?n=Moses.Overview`

(ii) transformer with additive visual conditioning based on VideoSum features (`Trans-AVC-VS`)

(iii) transformer with attention to visual features based on reshaped VideoSum features (`Trans-AVF-VS`)

(iv) transformer with attention to visual features based on convolutional layer output features (`Trans-AVF-CLO`)

(v) transformer with attention to visual features based on probability-scaled action category embedding features (`Trans-AVF-PSACE`)

See Chapters 3.2 and 2.2.4 for more details of the settings above.

All five model variants are trained and tested for three types of masked source texts: `ORG`, `ACT`, and `ALL`.

4.5.3 Training

For Phase 1, we train all the models on two Nvidia Tesla V100 GPUs (32GB memory each) with a patience of 10 epochs for early stopping. The batch size is 3072 per GPU hence 6144 in effect, and a base learning rate of 0.05 with 8,000 warm-up steps (following Ive et al. (2019)) is used for the Adam optimiser (Kingma and Ba, 2014). Since `tensor2tensor` 1.3.0 RC1 does not support validation on multiple GPUs during training, we disable automatic evaluation during training and instead use an Nvidia RTX 2080 Ti GPU to load each new checkpoint for inference on the validation set during training. Hence, we achieve on-the-fly validation based on the true `BLEU` scores (against the gold standard) obtained by the checkpoints, and stop the training when the early stopping criterion is met. We pick the checkpoint with the highest real `BLEU` score on the validation set for inference on the test set.

Our `BLEU` computation for checkpoint selection is done using the `t2t-bleu` script of `tensor2tensor` , comparing the outputs of the checkpoint models with the tokenised & lowercased Portuguese translations.

4.6 Phase 2: MMT with Transcripts

4.6.1 Preprocessing

Punctuation-Related

The transcripts from the speech recognition model are punctuation-free, therefore the question of whether and how to introduce punctuation was carefully considered.

We started by investigating the impact of punctuation on translation quality. Specifically, we eliminated punctuation from the original English subtitles $\mathcal{S}_o$ as $\mathcal{S}_{pr}$ (“pr” is short for punctuation-removed) and trained two vanilla transformers $\mathcal{T}_o^S$ and $\mathcal{T}_{pr}^S$ on $\mathcal{S}_o$ and $\mathcal{S}_{pr}$ respectively. As a result, we recorded a 56.64 BLEU achieved by $\mathcal{T}_o^S$, whereas the figure was 52.60 by $\mathcal{T}_{pr}^S$. Therefore, our conclusion was that punctuation, if correct, was considerably beneficial according to BLEU standards.

Hence, as the next step, punctuation introduction was explored. Popular punctuators, such as those with relatively many stars on GitHub, are mostly neural models trained on materials such as TED talks or website contents (e.g. Tilk and Alumäe (2016)). Mindful of the potential domain shift from the datasets used by those models to our transcripts, we trained our own vanilla-transformer punctuator P based on $\mathcal{S}_{pr}$ and $\mathcal{S}_o$ as a trial analysis, and then used the model to "translate" the original transcripts $\mathcal{TR}_o$ into their punctuated version $\mathcal{TR}_p$.

We then trained two more vanilla transformers for transcript-Portuguese translation: $\mathcal{T}_o^{TR}$ based on $\mathcal{TR}_o$ and $\mathcal{T}_p^{TR}$ based on $\mathcal{TR}_p$. Despite the minimal domain shift (i.e. from the subtitles to the transcripts), $\mathcal{T}_o^{TR}$ scored 40.30 BLEU on the validation set, whereas $\mathcal{T}_p^{TR}$ was only able to obtain 39.77. This is surprising, since the bias of P, caused by its training on the "subtitle domain" instead of the "transcript domain", was expected to rectify the "mistakes" made during the transcribing process (e.g. "A lot of these birds aren't tame" in subtitles, transcribed as "a lot of these birds are tame"), in addition to introducing punctuation, in terms of the ways in which the punctuator was assumed to be able to helpful for translation.

Upon closer inspection of $\mathcal{TR}_o$ and $\mathcal{TR}_p$, we noticed that P generally added natural-feeling punctuation marks but sometimes wrongly changed words in $\mathcal{TR}_o$. For example, the sentence "and you're here the shoulders aren't creeping up" in $\mathcal{TR}_o$ was punctuated into "and you're here , the shoulders aren't as up ." in $\mathcal{TR}_p$, and consequently the former was translated by $\mathcal{T}_o^{TR}$ as "e você está aqui , os ombros não estão subindo ." while $\mathcal{T}_p^{TR}$ produced "e você está aqui , os ombros não estão tão altos ." when given the latter, where "subindo" is correct and corresponds to "creeping up" whereas "tão altos" is in line with "as up". In other words, $\mathcal{T}_p^{TR}$ generated the accurate translation of a sentence that was incorrectly modified by P.

Thus, our conclusion was that the merits of P were outweighed by its mistakes. Since there were unlikely to be better punctuators due to domain shifts, we proceeded using the punctuation-free original transcripts $\mathcal{TR}_o$ for translation in Phase 2.

Vocabularies

As mentioned in the preceding section, we experimented with P which was trained on $\mathcal{S}_{pr}$ and $\mathcal{S}_o$. Since the vocabulary of $\mathcal{S}_o$ subsumes that of $\mathcal{S}_{pr}$ (the only extra tokens being the punctuation marks) and it is generally a good practice to train a punctuator with the same dictionary on the source and target sides, we use the same vocabulary for $\mathcal{S}_o$, $\mathcal{S}_{pr}$, $\mathcal{TR}_o$ and $\mathcal{TR}_p$.

Therefore, we again follow the "tokenisation → removing non-printing characters → replacing unicode punctuation marks with ASCII approximations" preprocessing pipeline for both $\mathcal{S}_o$ and $\mathcal{TR}_o$, then lowercase $\mathcal{S}_o$ ($\mathcal{TR}_o$ is already lowercased), and finally learn 20,000 BPE merge operations on the concatenation of the preprocessed

$\mathcal{S}_o$ and $\mathcal{TR}_o$, leading to a shared vocabulary of 19,300 subtokens. We use the same Portuguese vocabulary obtained in Chapter 4.5.1.

4.6.2 Models

For Phase 2, we involve all five types of transformers mentioned in Chapter 4.5 except that we use Ten-Hot Action Category Embeddings features for (v), making it `Trans-AVF-THACE`. Additionally, 10 types of deliberation networks participate. Therefore, the complete list of models we use in Phase 2 is as follows:

(i) `Trans-TO`

(ii) `Trans-AVC-VS`

(iii) `Trans-AVF-VS`

(iv) `Trans-AVF-CLO`

(v) `Trans-AVF-THACE`

(vi) text-only deliberation – additive deliberation at second pass (`Delib-TO-A`)

(vii) deliberation with additive visual conditioning based on VideoSum features – additive deliberation at second pass (`Delib-AVC-VS-A`)

(viii) deliberation with attention to visual features based on reshaped VideoSum features – additive deliberation at second pass (`Delib-AVF-VS-A`)

(ix) deliberation with attention to visual features based on convolutional layer output features – additive deliberation at second pass (`Delib-AVF-CLO-A`)

(x) deliberation with attention to visual features based on ten-hot action category embedding features – additive deliberation at second pass (`Delib-AVF-THACE-A`)

(xi) text-only deliberation – cascade deliberation at second pass (`Delib-TO-C`)

(xii) deliberation with additive visual conditioning based on VideoSum features – cascade deliberation at second pass (`Delib-AVC-VS-C`)

(xiii) deliberation with attention to visual features based on reshaped VideoSum features – cascade attention at second pass (`Delib-AVF-VS-C`)

(xiv) deliberation with attention to visual features based on convolutional layer output features – cascade attention at second pass (`Delib-AVF-CLO-C`)

(xv) deliberation with attention to visual features based on ten-hot action category embedding features – cascade attention at second pass (`Delib-AVF-THACE-C`)

See Chapters 3.2 and 2.2.5 for more details of the settings above.

As mentioned before, we only have one type of source text in Phase 2: How2 video transcripts, i.e. $\mathcal{TR}_o$.

It is worth noting that the multimodal deliberation networks in Ive et al. (2019) utilise the first 3 out of the total 6 decoder blocks at the second pass for additive attention to the first pass decoding output, which is referred to as "half deliberation" throughout the remainder of this report. As a preliminary investigation, we trained both `Delib-TO-A` and `Delib-TO-C` with both half and full deliberation, and found full deliberation to be generally worse-performing with a `BLEU` delta of as much as 1.5 compared to its half-deliberation counterpart. Therefore, we only use half deliberation for all our experiments involving the models listed above.

4.6.3 Source Augmentation for Deliberation

To obtain more material for training the second pass decoder in deliberation networks, we use the trick from Ive et al. (2019): generate the 10-beam first pass translation candidates for the training examples, so that the amount of training data becomes 10-fold, as each original example

(source, visual features, gold-standard)

is expanded into

(source, visual features, first-pass translation candidate #1, gold-standard),
(source, visual features, first-pass translation candidate #2, gold-standard),
. . .
(source, visual features, first-pass translation candidate #10, gold-standard)

We also experimented with 2- and 4-beam deliberation, the former used by the original paper on deliberation networks, before proceeding with 10-beam. Specifically, we trained `Delib-TO-A` with 2-beam, 4-beam and 10-beam deliberation separately, and found the performance on the validation set of the former two to be slightly lower than the latter (within 0.5 `BLEU`). Therefore, we ultimately decided to use 10-beam source augmentation for all the deliberation networks.

4.6.4 Training

Considering the nature of deliberation being refining a first-pass translation, we adopt the strategy from Ive et al. (2019): first train the underlying transformer model until convergence, and use their weights for initialising the encoder and first pass decoder of the deliberation model. We then train the network till its convergence.

A distinction from Ive et al. (2019), however, is that we freeze the encoder and first-pass decoder after they are imported from a trained transformer, so that the only part of a deliberation network that is trained in this project is the second pass decoder.

The primary reason for this choice is consistency. As previously mentioned, the first pass results that the second pass decoder attends to is a sequence of [first-pass pre-softmax hidden state; first-pass decoding result (token)] concatenations. For first-pass tokens, we directly use the output of the underlying transformer structure. If we fine-tune the first-pass decoder, it will mean that the two parts of the aforementioned concatenations will effectively come from two different decoders, which is inconsistent with the basic idea of deliberation.

Another justification is that we noticed that updating the second-pass decoder only resulted in significantly quicker convergence and also better performance. Specifically, we trained `Delib-TO-A` both updating the second-pass decoder only (partial update) and updating the whole model (full update), and found the parameter size of the former was 139,565,056 float32 units (4 bytes each) whereas the number was 374,506,496 for the latter. Therefore, partial update naturally leads to faster convergence. Surprisingly, a 1.5-`BLEU` drop was observed when full update was used compared to partial update. Therefore, we conducted our later Phase-2 experiments with partial update only.

For Phase 2, we train all the models on one Nvidia RTX 2080Ti (10GB memory each) with a batch size of 1024, a base learning rate of 0.02 with 8,000 warm-up steps (we tried 0.05 initially to be consistent with the transformer models and Ive et al. (2019), but it led to divergence during the training of `Delib-AVC-VS-A` and `Delib-AVC-VS-C`, so we decreased it to 0.02, which prevented the problem) for the Adam optimiser, and a patience of 3 epochs for early stopping based on `approx-BLEU`, a metric reported by `tensor2tensor` and generally reflective of the true BLEU scroes obtained by the model on the validation set. After the training terminates, we evaluate all the checkpoints on the validation set and compute the real `BLEU` scores of their outputs, based on which we select the best model for inference on the test set.

Like in Phase 1, Our `BLEU` computation for checkpoint selection is done using the `t2t-bleu` script of `tensor2tensor` , comparing the outputs of the checkpoint models with the tokenised & lowercased Portuguese translations. It is worth noting that we only keep the most recent up-to 100 checkpoints simultaneously.

4.6.5 Subsetting Test Set with Transcript Faithfulness

It was observed during preliminary experiments that transcribing quality varies across examples, where some transcribed sentences differ from the actual subtitles by only punctuation marks while some others completely changed meaning. Therefore, for more informative evaluation, we divide the test set into three subsets: faithful, moderately unfaithful, and highly unfaithful.

Specifically, we compare the tokenised and BPE-ed original transcripts (i.e. $\mathcal{TR}_o$) against the tokenised, lowercased, punctuation-removed and BPE-ed subtitles (i.e. $\mathcal{S}_{pr}$) sentence by sentence and calculate their Levenshtein distances (Levenshtein, 1966), and then normalise the result by the length of the transcript sentence, on the grounds that the normalised metric reflects the normalised effort of editing the transcript sentence into the true subtitle.

For the transcript sentences in the test set that have zero distance to their subtitle counterparts, we label them as “faithful”. On the other hand, those with a normalised distance larger than 0.25 are tagged “highly unfaithful”, and all the other sentences are marked as “moderately unfaithful”. This subsetting procedure leads to 527 faithful, 1,402 moderately unfaithful, and 376 highly unfaithful transcript sentences in the test set of 2,305 example in total. Our assumption is that visual information should be helpful for translating the unfaithful examples. We show below one (transcript, subtitle) pair for each faithfulness level (recovered from their preprocessed forms).

Faithful:
Subtitle: Today we’re going to be learning how to play Portal, a game by Valve Software.
Transcript: today we’re going to be learning how to play portal a game by valve software

Moderately Unfaithful:
Subtitle: So I am forcing the clay onto that part of my hand and here we go pushing it down, forcing it onto center like so.
Transcript: so i’m falseing the clay onto that pot of my hand and here we go pushing it down pull so get onto center like so

Highly Unfaithful:
Subtitle: DNA actually has a charge so that if you put in the right instrument the power source generates just the right amount of voltage.
Transcript: today i actually have a charge so to put it in the right as to our source to generates just to draw it around a bolster

Chapter 5

Results & Evaluation

5.1 Phase 1: MMT with Verb Masking

5.1.1 Scores

The BLEU (Papineni et al., 2002) scores[1] achieved by the five models in Phase 1 are shown in Table 5.1 .

Table 5.1: Results for the test set. We report BLEU scores. Bold highlights our best results. No multimodal system is significantly different (i.e. p-value ≤ 0.05) from its text-only counterpart (e.g. (Trans-AVF-CLO, ALL) compared to (Trans-TO, ALL)).

SETUP	ORG	ACT	ALL
TRANS-TO	55.9	53.6	44.1
TRANS-AVC-VS	55.6	53.6	44.2
TRANS-AVF-VS	55.7	53.3	44.0
TRANS-AVF-CLO	55.6	**53.8**	44.4
TRANS-AVF-PSACE	**56.2**	53.5	**44.5**

Trans-TO, our baseline, achieves a BLEU score of 55.9 for ORG. As expected, the baseline performs slightly (53.6 BLEU) and considerably worse (44.1 BLEU) as the masking progresses, which is consistent with the proportions of words masked in the latter two settings introduced in Chapter 4.2.1.

Trans-AVC-VS, exploiting visual features at the encoder side, performs on par with Trans-TO for ACT (difference within 0.1 BLEU) and ALL (+0.1 BLEU), but slightly worse for ORG, where a -0.3 BLEU delta is recorded.

One can see that Trans-AVF-VS, using the same visual features as Trans-AVC-VS but as a matrix on the decoder side, gives degraded performance in all three settings of

[1] We measure all our model performances with Multeval (Clark et al., 2011) throughout this thesis. We use tokenised and lowercased reference and hypotheses.

verb masking compared to the baseline: a 0.2 BLEU drop for ORG, 0.3 for ACT, and 0.1 for ALL. This suggests that the global visual features vector functions better when used in its original vectorial form than when it is artificially reshaped for attention from the decoder.

Trans-AVF-CLO, on the other hand, shows more improvements for ACT and ALL, with a 0.2 BLEU increase for the former compared to the baseline placing the model in the first place among the five for ACT. The jump for ALL is 0.3 BLEU, also evident.

Finally, Trans-AVF-PSACE enables a 0.3 BLEU improvement compared to the baseline for ORG and is thus our best model for the setting. It is worth noting that Trans-AVF-PSACE is also the only multimodal model that beats our Trans-TO baseline for ORG. For ACT, the model fares worse than the baseline with a 0.1 BLEU drop. For ALL, however, it again shows a major improvement of 0.4 BLEU over Trans-TO.

Overall, in terms of the BLEU automatic metric results, our multimodal models that exploit VideoSum features are on par with, if not lagging behind, the text-only baseline. Trans-AVF-CLO and Trans-AVF-PSACE, which utilise convolutinal and word embedding features respectively, generally fare better than the baseline in more than one scenarios, proving the benefits of using richer visual information for multimodal translation.

We note that, interestingly, no multimodal model for ACT is able to achieve a score similar to Trans-TO for ORG, which is especially interesting for Trans-AVF-CLO and Trans-AVF-PSACE, since their convolutional and word embedding features come directly from a CNN that classifies videos into action categories whose labels correspond to the action verbs masked in ACT. This is indication that Trans-AVF-CLO and Trans-AVF-PSACE have not fully exploited the visual features or that their visual features may not have captured the elements that are key to translation.

One can also see that the score gap between ORG and ACT is smaller for some multimodal models than it is for the baseline (2.3 BLEU): Trans-AVC-VS and Trans-AVF-PSACE shrink the gap by 0.3 and 0.5 BLEU points respectively. Similarly, Trans-AVC-VS and Trans-AVF-CLO successfully shrink the gap between ORG and ALL by 0.4 and 0.6, which are indeed pronounced differences.

5.1.2 Incongruence Analysis

It is clear from Table 5.1 that the improvements over the Trans-TO baseline achieved by the multimodal models are generally modest, and indeed the baseline performs even better in some settings. This leads to the fundamental question: does multimodality help translation at all?

To find the answer, we follow the incongruent decoding approach proposed by Caglayan et al. (2019a), where our multimodal models are fed with mismatchd visual features, and the score difference between the normal and incongruent decoding results should be telling of how much visual features matter to the models.

The general assumption is that a model will have learned to exploit visual information to help with its translation, if it shows substantial performance degradation when it is given wrong visual features.

We carry out the incongruence test on the test set of 2,305 examples, and feed the visual features to the models in reverse order to ensure the visual features are incorrect. The score details are in Table 5.2.

Table 5.2: Results for the test set, **with incongruence**. We report the BLEU score changes w.r.t. the congruent-decoding counterpart systems in Table 5.1. † marks incongruent decoding results that are significantly different (p-value ≤ 0.05) from their congruent-decoding counterparts

SETUP	ORG	ACT	ALL
TRANS-AVC-VS	↑ 0.1	↓ 0.7 †	↓ 1.0 †
TRANS-AVF-VS	↓ 0.1	↓ 0.3	↓ 0.5 †
TRANS-AVF-CLO	↓ 0.3	↓ 0.5 †	↓ 0.8 †
TRANS-AVF-PSACE	↓ 0.1	↓ 0.4 †	↓ 0.3

Immediately clear from the table is the fact that incongruent visual features lead to degraded performance in all but one scenarios (Trans-AVF-VS for ACT), which is proof that the multimodal models have learned to utilise visual information to aid translation.

Upon closer inspection, one will notice that the deltas are relatively small for the ORG setting: Trans-AVF-CLO shows a 0.3 BLEU decrease, Trans-AVF-VS and Trans-AVF-PSACE both suffer a drop of 0.1 BLEU, whereas Trans-AVC-VS gains 0.1 BLEU. For ACT, however, the score differences are all negative and considerably larger: -0.7, -0.3, -0.5 and -0.4 for Trans-AVC-VS, Trans-AVF-VS, Trans-AVF-CLO, and Trans-AVF-PSACE respectively. The figures are even more pronounced for ALL: -1.0, -0.5, -0.8 and -0.3 for the mutimodal models.

Those numbers indicate the more incomplete the source text is the more important visual features are to the multimodal models. A mask-free sentence (e.g. ORG) may not need visual information to help with translation, while the same information can be crucial in scenarios where elements present in the visual domain but missing from the source sentence (e.g. ACT) can be recovered and correctly represented in the translation by a multimodal model.

5.1.3 Human Analysis

Over the years, it has been pointed out in multimodal machine translation research that automatic metrics such as BLEU can be an imperfect lens through which translation quality is determined, since nuances in translations that are signs of good inference quality according to human judgement may not be well represented by automatic metrics, such as the translation subtleties for which multimodality is helpful (Elliott et al., 2017b; Barrault et al., 2018b).

Therefore, for Phase 1, four native Portuguese speakers who are also fluent in English were invited to assess the inference results of three models. Specifically, Trans-TO, Trans-AVF-CLO, and Trans-AVF-PSACE produced their outputs as our candidate translations for ACT, and the four annotators were each given the same 50 randomly selected examples from the test set for which the three models had different outputs from one another. The Portuguese reference translations for those examples were also given to the annotators, who were asked to rank the three candidate translations on a scale of 1 to 3 while allowing ties (Bojar et al., 2017). Also, three-zero ranks were also permitted for the candidate translations if they were all considered too low-quality for a judgement. Additionally, the annotators were also asked to take into account not only the sentence-level translation quality, but also how well the ACT-masked verbs in the source sentence.

We adopt the Ratio of Wins and Ties strategy (Callison-Burch et al., 2011) to process the scores. Specifically, since we have for each example a ranking among the three systems, we can count the number of times a system is better, worse or equal to another system, and then the score of each system becomes the proportion of times that it defeats or forms a tie with another system. Table 5.3 shows our human evaluation scores computed in this way.

Trans-TO	Trans-AVF-CLO	Trans-AVF-PSACE
0.75	0.73	0.81

Table 5.3: Human ranking results for ACT: micro-averaged over four annotators.

In contrast to the BLEU scores given to the three systems shown in Chapter 5.1.1, human annotators in general favoured Trans-AVF-PSACE which has a BLEU of 53.5, the lowest among the three. On the other hand, Trans-AVF-CLO, the system favoured by BLEU, had the least support from the annotators. See Figure 5.1 for some examples where Trans-AVF-CLO or Trans-AVF-PSACE (i.e. one of the two multimodal models) beats Trans-TO baseline (text-only).

Although the human evaluation results demonstrate that Trans-AVF-PSACE performs better than the verdict given by BLEU, this noticeable discrepancy between BLEU and human judgement warrants more investigation, which we leave for future work.

EN	So, how do I make sure that I spin all the way around, or, how do you make sure?
Trans-TO	Então, como eu me certifico de cortar a toda a volta, ou como você se certifica?
Trans-AVF-CLO	Então, como eu me certifico de dar a volta, ou, como você se certifica?
Trans-AVF-PSACE	Então, como eu me certifico de que eu viro todo o caminho, ou, como você se certifica?
PT	Então, como eu me certifico de girar ao redor, ou, como você se certifica?

(a) Trans-AVF-CLO guesses the masked word spin correctly as *dar a volta*, while the Trans-TO model translates it incorrectly as *cortar* (cut) and Trans-AVF-PSACE translates it partially correctly as *virar* (turn)

EN	In this clip we're talking about footwork, we're going to be covering the moving forward aspect of it.
Trans-TO	Neste clipe, estamos falando de trabalho de pés, vamos discutir o aspecto da frente dele.
Trans-AVF-CLO	Neste clipe, estamos falando sobre o trabalho de pés, vamos cobrir o aspecto da mudança para a frente.
Trans-AVF-PSACE	Neste clipe, estamos falando de footwork, vamos discutir o aspecto do movimento em movimento.
PT	Neste pequeno vídeo, estamos falando de trabalho de pés, vamos estar cobrindo o aspecto avançado.

(b) Trans-AVF-CLO guesses talking and covering correctly as *falar* (talk) and *cobrir* (cover); the other models get the first word right, but translate the other word as *discutir* (discuss)

EN	I might use the sixty degree wedge a bit too, but the sand wedge obviously is useful for getting out of the ruff, hitting the ball from the fairway, getting out of sand.
Trans-TO	Eu poderia usar a cunha de sessenta graus também, mas a cunha de areia, obviamente, é útil para sair do ruff, tirar a bola do fairway, sair da areia.
Trans-AVF-CLO	Eu poderia usar a cunha de sessenta graus um pouco também, mas a cunha de areia obviamente é útil para sair do pescoço, bater na bola do fairway, saindo da areia.
Trans-AVF-PSACE	Eu posso usar a cunha de sessenta graus um pouco também, mas a cunha de areia obviamente é útil para sair do ruff, bater a bola do fairway, sair da areia.
PT	Eu também poderia usar a wedge de sessenta graus, mas a sand-wedge, obviamente, é til para sair do ruff, acertar a bola no fairway, sair da areia.

(c) Trans-AVF-CLO and Trans-AVF-PSACE guess hit correctly as *bater* (the gold-standard *acertar* has the same meaning this example), while the *text-only* model translates it as *tirar* (remove)

Figure 5.1: Examples of improvements of Trans-AVF-CLO and Trans-AVF-PSACE over the text-only baseline. Underlined text denotes masked words and their translations.

5.2 Phase 2: MMT with Transcripts

5.2.1 Scores

The `BLEU` scores achieved by each model detailed in Chapter 4.6.2 under each faithfulness setting defined in Chapter 4.6.5 are shown in Table 5.4, where the "overall" column lists the performances measured on the entire test set (2305 examples) regardless of faithfulness.

	Setup	Overall	Faithful	Moderately Unfaithful	Highly Unfaithful
`Trans`	`Trans-TO`	**40.3**	58.1	40.4	22.2
	`Trans-AVC-VS`	40.2	57.7	40.4	21.5
	`Trans-AVF-VS`	40.2	**58.6**	40.1	22.7
	`Trans-AVF-THACE`	**40.3**	57.2	**40.5**	22.1
	`Trans-AVF-CLO`	**40.3**	**58.6**	40.1	**22.9**
`Delib-A`	`Delib-TO-A`	38.2	55.3	38.1	**21.4**
	`Delib-AVC-VS-A`	36.7 †	54.3	36.5 †	19.3 †
	`Delib-AVF-VS-A`	**38.3**	55.3	**38.2**	21.2
	`Delib-AVF-THACE-A`	37.5 †	55.4	37.5	20.0 †
	`Delib-AVF-CLO-A`	37.8	**55.8**	37.8	20.3
`Delib-C`	`Delib-TO-C`	37.0	54.3	36.7	**21.1**
	`Delib-AVC-VS-C`	36.9	**55.3**	36.6	19.7
	`Delib-AVF-VS-C`	**38.0** †	**55.3**	**38.0** †	20.9
	`Delib-AVF-THACE-C`	37.9 †	55.0	37.8 †	21.0
	`Delib-AVF-CLO-C`	37.6	54.0	37.5 †	21.0

Table 5.4: `BLEU` scores of all the transformer and deliberation models under all the transcript settings. Bold highlights our best system in each (architecture, transcript) setting, and † indicates significant difference (p-value ≤ 0.05) compared to the corresponding baseline, e.g. (`Trans-AVF-THACE`, moderately unfaithful) compared to (`Trans-TO`, moderately unfaithful)

One immediate observation is that the translation quality of every model deteriorates as the transcript faithfulness decreases. From completely faithful to moderately unfaithful, a `BLEU` gap of 17 to 18 is generally resulted. A similar amount of score drop is also presented as the transcripts progress from moderately to highly unfaithful. Also, one will notice that the model performances under the moderately unfaithful setting are similar to their overall ones, which indicates that the transcripts are, on average, on the "moderately unfaithful" level defined previously, also justifying our threshold choice (i.e. positive normalised Levenshtein distance below 0.25 for being moderately unfaithful).

A surprising result is that the transformer-based models outperform their deliberation counterparts in every transcript setting, whether they are text-only or multimodal. Specifically, Trans-TO overall (40.3) beats Delib-TO-A (38.2) by 2.1 BLEU and Delib-TO-C (37.0) by 3.3, which are considerable score differences. This performance degradation directly contradicts the improvement by deliberation over the transformer shown by Ive et al. (2019), where Delib-TO-A surpasses Trans-TO by as much as 1.3 BLEU. It is improbable that our approach is fundamentally flawed, since we trained Delib-TO-A and Trans-TO on Multi30K which is used by Ive et al. (2019) and found similar improvements. On the other hand, our text pre- and post-processing is equally unlikely to be unsound, as we checked repeatedly and also due to the fact that our Trans-TO shows no abnormality. As for the system output of the deliberation models, we did not observe any obvious abnormalities common in machine translation, as was confirmed by a native Portuguese speaker who is also an MT expert.

For the transformers, one will notice that, interestingly, Trans-TO is on par with Trans-AVF-THACE and Trans-AVF-CLO as one of the overall top-performing models, differing from the latter two within 0.1 BLEU, despite not taking the lead for any faithfulness level — Trans-AVF-CLO and Trans-AVF-VS are the best (58.6 BLEU) for faithful transcripts, Trans-AVF-THACE is in the first place (40.5) for moderately unfaithful ones, and Trans-AVF-CLO tops the list for the highly unfaithful subset. This can only be explained by a statistical advantage — Trans-TO is a close second (40.4 BLEU) for moderately unfaithful sentences which constitute more than 60% (1,402 out of 2,305) of the dataset, whereas Trans-AVF-CLO, beating Trans-TO by 0.5 BLEU for faithful examples and 0.7 BLEU for highly unfaithful (which are substantial improvements in MT), is among the worst-performing model (40.1 BLEU, lagging behind Trans-TO by 0.3) for moderately unfaithful examples.

Comparing the ORG column of Table 5.1 and the Overall column of Table 5.4, one can conclude that visual information in Phase 2 makes statistically less difference for the transformer-based models — Trans-AVC-VS lags behind Trans-TO by 0.1 BLEU compared to the 0.3 in Phase 1, Trans-AVF-VS trails Trans-TO by 0.1 BLEU in contrast to the 0.2, and Trans-AVF-CLO is practically the same as Trans-AVF-CLO in Phase 2 but the delta is -0.3 in Phase 1. Granted, there are certain configuration- and resources-related minor training differences between Phases 1 and 2, but the statistical closeness between the system scores should come down to the fundamental distinction: the source. It is probable that the transcripts, as degraded subtitles, preclude more accurate translation and, as a byproduct, bring different models closer together.

Following the idea of source corruption, one will notice, by comparing the ACT column of Table 5.1 and the Overall column of Table 5.4, that multimodality is also less statistically helpful on average, which is best illustrated by the 0.2 BLEU lead (53.8 vs. 53.6) Trans-AVF-CLO has over Trans-TO in Phase 1 in contrast to the virtually no difference in Phase 2. Since transcribing can be interpreted as introducing random noise as opposed to the deterministic action verb masking in ACT in Phase 1, it can be assumed that the same action convlutional features (i.e. CLO) are more useful in Phase 1 where they can more directly help fill the deliberately replaced verbs due to the close connection between the features and the verbs.

Additive deliberation paints a very different picture. The first difference is that Delib-AVF-VS-A, a multimodal model, now beats the baseline by 0.1 BLEU overall. The other multimodal additive deliberation networks, however, lag behind Delib-AVF-VS-A and Trans-TO significantly with deltas as large as 1.5 BLEU (Delib-AVF-VS-A) compared to Trans-TO. This is in stark contrast to the transformers, where the unimodal and multimodal systems perform similarly overall. Also different from the transformers is that Delib-TO-A, a text-only model, is the champion for the highly unfaithful subset, leaving the second best (Delib-AVF-VS-A) 0.2 BLEU behind. The distinction is particularly pronounced for models using CLO features: Trans-AVF-CLO (22.9) surpasses Trans-TO (22.2) by 0.7 BLEU on this subset, whereas Delib-AVF-CLO-A (20.3) is defeated by Delib-TO-A (21.4) by 1.1 BLEU. For THACE features, similarly, Delib-AVF-THACE-A is outperformed by Delib-TO-A by 0.6 BLEU on the moderately unfaithful examples, despite Trans-AVF-THACE being ranked the best in the same category.

There are also interesting parallels. For example, Delib-AVF-VS-A secures its overall performance through its strong results (38.2 BLEU, first place) on the majority subset — moderately unfaithful examples, not unlike Trans-TO and Trans-AVF-THACE. Also, CLO features again contribute to the winner model Delib-AVF-CLO-A for the faithful subset, outperforming its baseline (Delib-TO-A) by 0.5 BLEU, the same amount as the delta between Trans-AVF-CLO and Trans-TO.

Cascade deliberation yields some results similar to additive deliberation. Delib-AVF-VS-C (38.0 BLEU) is overall the best cascade-deliberation model just as Delib-AVF-VS-A is for additive deliberation, also thanks to its dominance (36.7 BLEU) on the moderately unfaithful subset. Again, Delib-TO-C becomes the winner (21.1 BLEU) on the highly unfaithful examples, but this time in spite of its considerably weaker overall results: second last overall (37.0 BLEU) and on the faithful (54.3 BLEU) and moderately faithful (36.7) subsets. Noticeably, Delib-TO-C is substantially worse BLEU-wise compared to Delib-TO-A with deltas as large as -1.4 on moderately unfaithful examples, which suggests the choice of deliberation mechanism indeed makes a difference.

A general observation can be made about the multimodal deliberation networks: apart from Delib-AVF-VS-C, all the cascade deliberation networks are able to beat the their baseline (i.e. Delib-TO-C) by a large delta, especially for Delib-AVF-VS-C (38.0 BLEU) and Delib-AVF-CLO-C (37.9 BLEU) which both surpass the baseline by close to 1 BLEU. The multimodal advantage is much less obvious the case, however, for additive deliberation, where AVF and AVC models struggle to compete with the 38.0-BLEU baseline (Delib-TO-A). Also, compared to the pronounced overall score difference between Delib-TO-A and Delib-TO-C, the multimodal additive-deliberation systems have much more similar scores with their cascade-deliberation counterparts.

Finally, the transformers and deliberation networks deliver a mixed verdict on the assumption we make in Chapter 4.6.5 that visual information can be more helpful for more unfaithful source sentences since the former may be able to help "correct" the semantic meaning of the latter. What we see in Table 5.4 is that the assumption clearly holds for two multimodal transformers: Trans-AVF-VS and Trans-AVF-CLO which beat their Trans-TO baseline by 0.5 and 0.7 BLEU respectively. For deliberation, however, Delib-TO-A and Delib-TO-C are the best in their model groups, contradicting the assumption. We investigate this further in Chapter 5.2.2

5.2.2 Incongruence Analysis

Similar to in Chapter 5.1.2, we carry out the same incongruent decoding with reverse-order visual features, and the results are shown in Table 5.5. We inspect the results of transformer-based, additive-deliberation-based, and cascade-deliberation-based models separately.

	Setup	Overall	Faithful	Moderately Unfaithful	Highly Unfaithful
Trans	Trans-AVC-VS	↓ 0.4 †	↓ 1.0	↓ 0.3	↓ 0.5
	Trans-AVF-VS	↓ 0.3	↓ 0.9 †	↓ 0.2	-
	Trans-AVF-THACE	↓ 0.5 †	↓ 0.4	↓ 0.3	↓ 1.0 †
	Trans-AVF-CLO	↓ 0.1	↓ 1.4 †	↑ 0.1	↑ 0.4
Delib-A	Delib-AVC-VS-A	↑ 0.1	↓ 1.0	↑ 0.3	↑ 0.5
	Delib-AVF-VS-A	-	-	-	↑ 0.2
	Delib-AVF-THACE-A	↑ 0.2	-	↑ 0.1	↑ 0.5
	Delib-AVF-CLO-A	↓ 0.1	↓ 0.7	↓ 0.2	↑ 0.2
Delib-C	Delib-AVC-VS-C	↓ 0.6 †	↓ 2.0 †	↓ 0.5	↑ 0.1
	Delib-AVF-VS-C	-	↓ 0.3	-	↓ 0.1
	Delib-AVF-THACE-C	↓ 0.2 †	↓ 0.3	↓ 0.2	-
	Delib-AVF-CLO-C	↓ 0.2	↓ 0.5	↑ 0.1	↓ 0.5

Table 5.5: BLEU score deltas caused by incongruent decoding, where "-" means insignificant change (i.e. within 0.1 BLEU) and † marks incongruent decoding results that are significantly different (p-value ≤ 0.05) from their congruent-decoding counterparts

For the multimodal transformers, the effect of incongruence is obvious, with all the models experiencing overall BLEU drops, of which the largest is 0.4 (Trans-AVC VS) and the smallest 0.1 (Trans-AVF-CLO). By inspecting the incongruence impact on translating the differently faithful subsets, we will notice that the largest drops tend to be from decoding the faithful transcripts, including a 1.4 BLEU decrease suffered by Trans-AVF-CLO. For moderately and highly unfaithful examples, on the other hand, the score changes with incongruence are not universally for the worse. In particular, Trans-AVF-CLO gets a boost of 0.1 and 0.4 from incongurence on the moderately and highly faithful examples respectively. It is also clear that score changes on the moderately unfaithful generally vary less, as the largest of them is -0.3 (Trans-AVC-VS). In all, those negative BLEU deltas caused by incongruence are indicative of the multimodal transformers utilising the visual features to help with their translation, especially so for faithful transcripts.

In stark contrast, additive deliberation with incongruent decoding leads to small overall BLEU increases for Delib-AVC-VS-A (+0.1) and Delib-AVF-THACE-A (+0.2), virtually no difference for Delib-AVF-THACE-A, and a slight decline (-0.1) for Delib-AVF-CLO-A. While the score changes are either negative (Delib-AVC-VS-A, Delib-AVF-CLO-A) or negligible (Delib-AVF-VS-A and Delib-AVF-THACE-A) on the faithful transcript sentences, they are universally positive on the highly unfaithful examples for the multimodal models. This suggests that correct visual information is not as useful for the additive-delib models as it is for the transformers, and in fact does harm to the performance of those multimodal additive-delib networks on those sentences.

Incongruence with cascade deliberation behaves more similar to the transformer case. Again, faithful examples all suffer translation quality degradation from 0.3 to 0.5 BLEU with the AVF models and a striking 2.0 BLEU with Delib-AVC-VS-C. Moderately and highly unfaithful subsets generally experience small BLEU deltas ranging from -0.2 to 0.1 with incongruent decoding of the multimodal models, except Delib-AVC-VS-C on the moderately unfaithful subset and Delib-AVF-CLO-C on the highly unfaithful one, both of which are a 0.5 BLEU decrease. Therefore, visual information is largely a positive force for multimodal cascade deliberation.

Across models, the determination can be made that transformer- and cascade-deliberation-based multimodal systems in general have evidently deteriorated performances led to by incongruent decoding, which is proof that correct visual information matters to those models and aids their decoding. Additive deliberaion on the other hand shows less reliance on and lower efficiency in utilising the visual features, showing minor overall BLEU drops and even boosts in the highly unfaithful scenario.

Interestingly, how much the visual modality matters (in other words, how significant the performance degradation is) is not necessarily correlated with the BLEU of the congruent decoding results. For example, Delib-AVF-CLO-C beats Delib-AVC-VS-C by 0.7 BLEU with normal decoding, but the former only suffers a 0.2-BLEU loss with incongruence whereas the figure for the latter is 0.6. The same can be said about Trans-AVF-CLO (-0.1 BLEU) and Trans-AVC-VS (-0.4 BLEU). This means that some multimodal models that are sensitive to incongruence likely complement visual attention with textual attention but without getting better output relying on the visual modality. We leave it to future work for further investigating this phenomenon.

On the macro level, we can see that whichever the multimodal architecture, be it transformer-based or deliberation-based, it is very clear that visual features are most helpful on faithful examples, based on how much damage incongruence causes. For highly unfaithful transcript sentences, however, it is a lot less certain, especially considering the BLEU jumps in additive deliberation. Hence, we can answer the previous question: No, our action visual features are not more helpful on more unfaithful examples, but instead on more faithful ones. This is in contradiction to the conclusions we drew from the incongruence results of Phase 1, where ALL, which can be considered "more unfaithful" than ORG and ACT, suffered the largest loss of translation quality. We do not have a definite answer that reconciles those facts, but one plausible explanation is that our action-based visual features are more suited for recovering from targeted verb masking than from source corruption caused by the random noise of the transcribing process.

5.2.3 Attention Visualisation

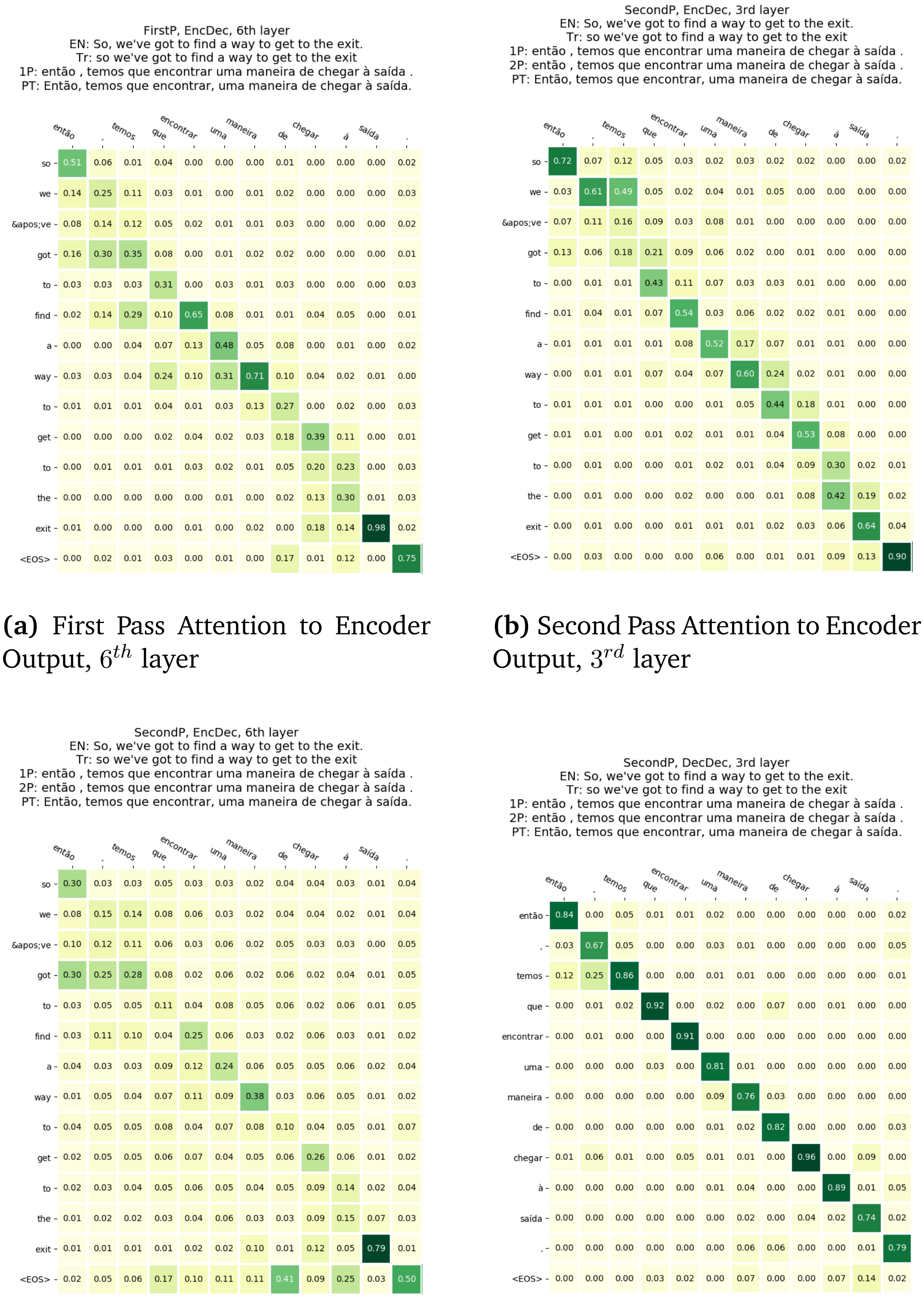
To better understand our models, we visualise textual and visual attention on test-set examples.

Textual Cross Attention with Deliberation

As mentioned in Chapter 5.1.1, the deliberation networks deliver worse performance compared to their transformer counterparts. Considering the half-deliberation nature of those networks (i.e. second-pass decoder attention to first-pass results only at the first three layers. See Chapter 4.6.2 for more details), we extract the textual cross attention paid to the encoder output (English) by `Delib-TO-A` and `Delib-TO-C` at the 6^{th} layer of the their first pass decoders and the 3^{rd} and 6^{th} layers of their second pass decoders, as well as their attention to the first pass output (Portuguese), with an example shown in Figures 5.2 and 5.3. We average the attention weights at the 16 heads for the illustration. Note that Figures 5.3a and 5.3a are identical, since `Delib-TO-A` and `Delib-TO-C` rely on the same un-fine-tuned encoder and first pass decoder from `Trans-TO`.

Despite their different ways of integrating second-pass textual attention to the encoder output and first pass [hidden state, decoding result] concatenations, we can see that `Delib-TO-A` and `Delib-TO-C` have similar patterns. In both models, the shared first pass decoder exhibits appropriate diagonal-style lexical correspondence between the English words and their Portuguese counterparts, such as find–encontrar and exit-saída, and the same can be said about the middle (i.e. 3^{rd}) layer of the second pass decoder. For the second-pass attention to the first pass results, shown in Figures 5.2d and Figures 5.3d, the correlation is much stronger, as manifested by the same-word attention all the way from então-então to saída-saída.

However, a problem existing in both models is also revealed in Figures 5.2c and 5.3c: much less focused attention to the encoder output at the last (6^{th}) second pass layer compared to at the middle (i.e. 3^{rd}) one. This is a direct contradiction of the received wisdom about layered attention structures in general that layers higher up tend to have learned to concentrate on the “useful” or “right” words for translation and therefore show more focused attention than the previous layers. We believe that this irregularity is directly responsible for the degraded performance of the deliberation networks.



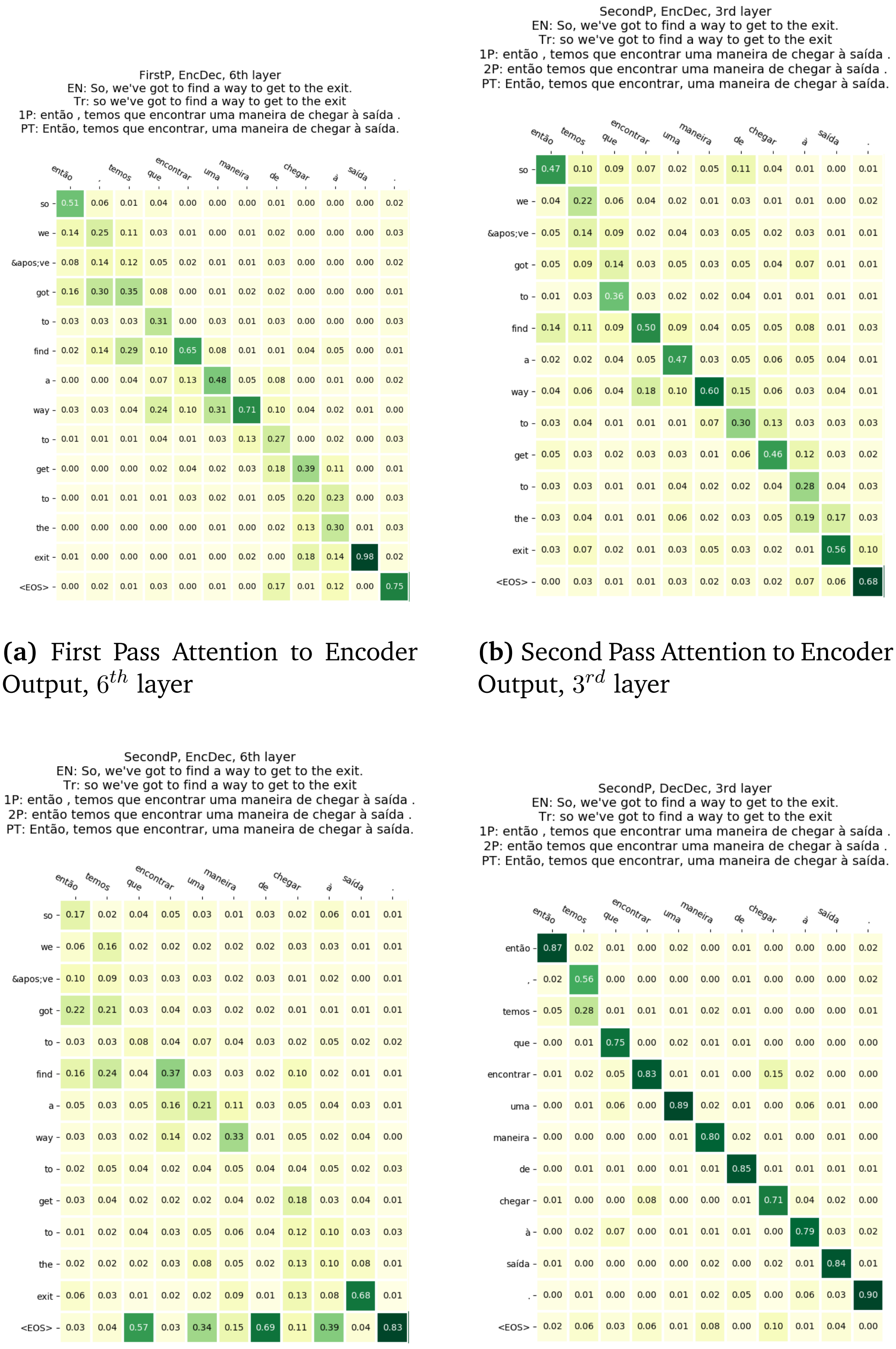
(a) First Pass Attention to Encoder Output, 6^{th} layer

(b) Second Pass Attention to Encoder Output, 3^{rd} layer

(c) Second Pass Attention to Encoder Output, 6^{th} layer

(d) Second Pass Attention to First Pass Output, 3^{rd} layer

Figure 5.2: Textual attention of `Delib-TO-A` at 4 different layers of first-pass and second-pass decoders. EN: Original Subtitle; Tr: Transcript Sentence; 1P: First Pass Decoding Output; 2P: Second Pass Decoding Output; PT: Portuguese Reference



(a) First Pass Attention to Encoder Output, 6^{th} layer

(b) Second Pass Attention to Encoder Output, 3^{rd} layer

(c) Second Pass Attention to Encoder Output, 6^{th} layer

(d) Second Pass Attention to First Pass Output, 3^{rd} layer

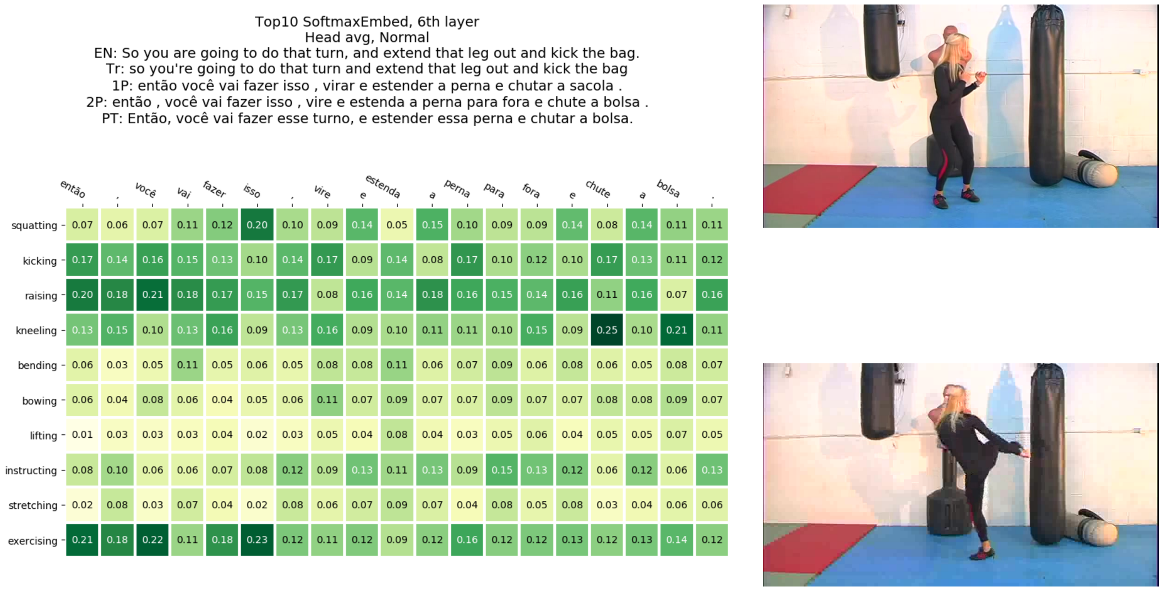
Figure 5.3: Textual attention of `Delib-TO-C` at 4 different layers of first-pass and second-pass decoders. EN: Original Subtitle; Tr: Transcript Sentence; 1P: First Pass Decoding Output; 2P: Second Pass Decoding Output; PT: Portuguese Reference

Visual Attention: Ten-Hot Action Category Embedding

For each THACE-based example, we show two of the 16 equi-distant frames of a video which were fed to the action recognition network that generated the visual features, in order to make it possible for the reader to get the gist of the video.

The general observation about visual attention by the transformer and deliberation networks to ten-hot action category embeddings is that, when the video frames are closely associated with the detected actions, the averaged head attention (16 heads in total) is mostly distributed across the categories that are most relevant, no matter the word that is being generated.

An example of the observation above is given in Figure 5.4 where the video segment demonstrates a turn-and-kick movement. The Delib-AVF-THACE-A model on average visibly focuses its attention on four categories: “kicking”, “raising”, “kneeling” and “exercising”. Among them, “kicking” and “exercising” are obvious elements present in the video, while “raising” and “kneeling” are visually similar actions. Since the action category embeddings are not weighted but instead in their original forms, we can see that the model has learned to focus on the relevant actions. It is also clear that the model has a relatively narrow range for attention — when generating Portuguese words such as “então” (“so”) and “fazer” (“do”), it focuses on the same relevant action categories as it does for the other more action-related words.

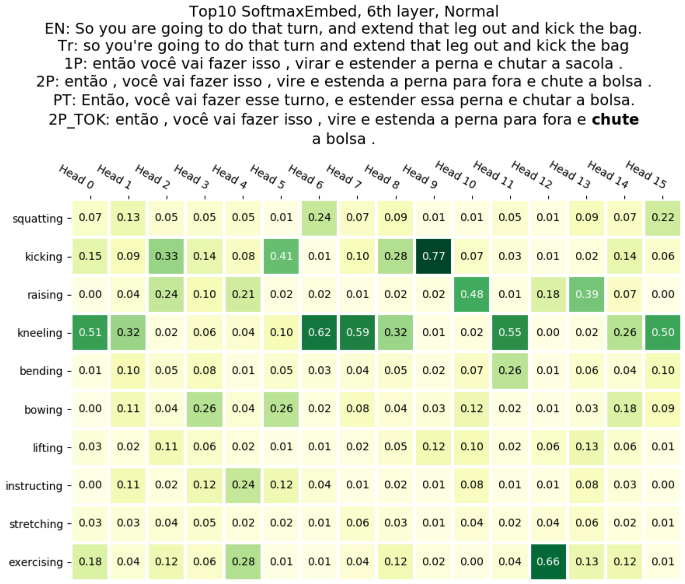


Top10 SoftmaxEmbed, 6th layer
Head avg, Normal
EN: So you are going to do that turn, and extend that leg out and kick the bag.
Tr: so you're going to do that turn and extend that leg out and kick the bag
1P: então você vai fazer isso , virar e estender a perna e chutar a sacola .
2P: então , você vai fazer isso , vire e estenda a perna para fora e chute a bolsa .
PT: Então, você vai fazer esse turno, e estender essa perna e chutar a bolsa.

	então	,	você	vai	fazer	isso	,	vire	e	estenda	a	perna	para	fora	e	chute	a	bolsa	.
squatting	0.07	0.06	0.07	0.11	0.12	0.20	0.10	0.09	0.14	0.05	0.15	0.10	0.09	0.09	0.14	0.08	0.14	0.11	0.11
kicking	0.17	0.14	0.16	0.15	0.13	0.10	0.14	0.17	0.09	0.14	0.08	0.17	0.10	0.12	0.10	0.17	0.13	0.11	0.12
raising	0.20	0.18	0.21	0.18	0.17	0.15	0.17	0.08	0.16	0.14	0.18	0.16	0.15	0.14	0.16	0.11	0.16	0.07	0.16
kneeling	0.13	0.15	0.10	0.13	0.16	0.09	0.13	0.16	0.09	0.10	0.11	0.11	0.10	0.15	0.09	0.25	0.10	0.21	0.11
bending	0.06	0.03	0.05	0.11	0.05	0.06	0.05	0.08	0.08	0.11	0.06	0.07	0.09	0.06	0.08	0.06	0.05	0.08	0.07
bowing	0.06	0.04	0.08	0.06	0.04	0.05	0.06	0.11	0.07	0.09	0.07	0.07	0.09	0.07	0.07	0.08	0.08	0.09	0.07
lifting	0.01	0.03	0.03	0.03	0.04	0.02	0.03	0.05	0.04	0.08	0.04	0.03	0.05	0.06	0.04	0.05	0.05	0.07	0.05
instructing	0.08	0.10	0.06	0.06	0.07	0.08	0.12	0.09	0.13	0.11	0.13	0.09	0.15	0.13	0.12	0.06	0.12	0.06	0.13
stretching	0.02	0.08	0.03	0.07	0.04	0.02	0.08	0.06	0.07	0.09	0.07	0.04	0.08	0.05	0.08	0.03	0.04	0.06	0.06
exercising	0.21	0.18	0.22	0.11	0.18	0.23	0.12	0.11	0.12	0.09	0.12	0.16	0.12	0.12	0.13	0.12	0.13	0.14	0.12

Figure 5.4: 1^{st} and 8^{th} frames of a video segment & Average-head attention of the second pass decoder of Delib-AVF-THACE-A to embeddings of the top 10 THACE categories detected from the same video segment. EN: original subtitle; Tr: transcript sentence; 1P: first-pass decoding output; 2P: second-pass decoding output; PT: Portuguese reference; 2P_TOK: BPE-ed & tokenised form of 2P

Interestingly, even when the action category label is presented in its lexical form in the sentence to be translated, the model does not necessarily pay the most attention to that category. In this same example, one will notice in Figure 5.4 that the model on average is 17% focused on “kicking” when generating the word “chute” (i.e. “kick” in Portuguese) in contrast to 25% on “kneeling”. If we inspect the attention of all the heads for “chute” in Figure 5.5, we will see that Head 8 is predominantly (77%) “kicking”-attending, and Heads 2 and 5, too, are mostly focused on “kicking”. This means the 16 heads have learned diverse attention schemes, and therefore the relevant or “correct/interesting” categories may not have the most attention on average but usually have some heads dealing with them.

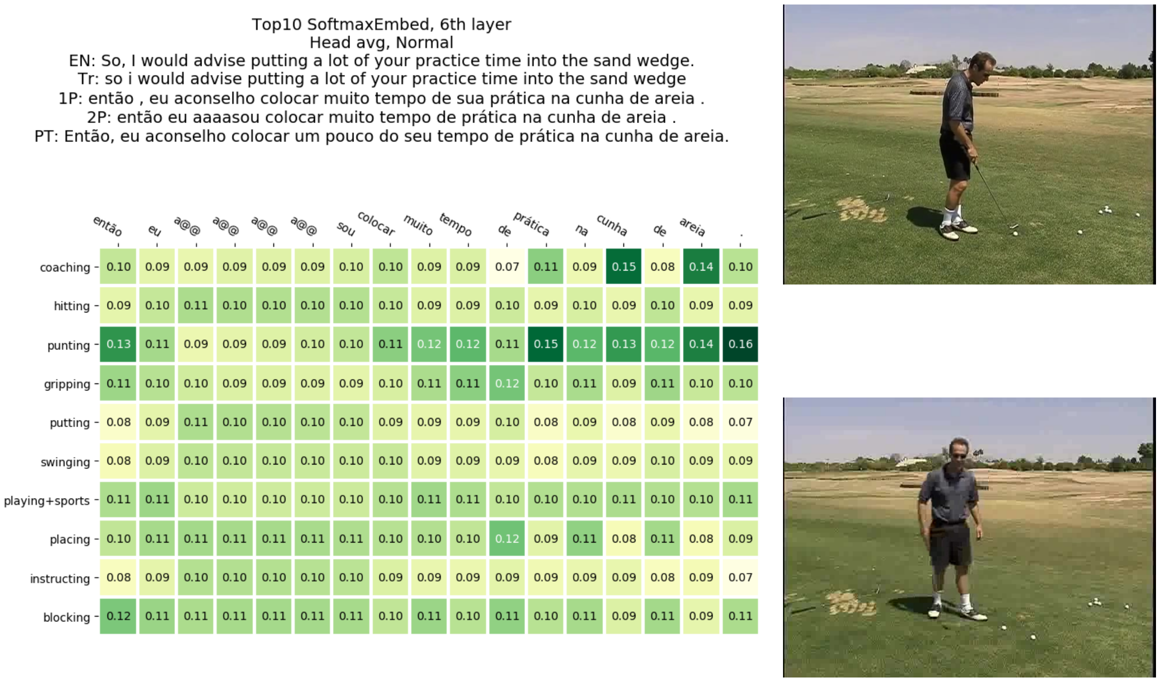


Top10 SoftmaxEmbed, 6th layer, Normal
EN: So you are going to do that turn, and extend that leg out and kick the bag.
Tr: so you're going to do that turn and extend that leg out and kick the bag
1P: então você vai fazer isso , virar e estender a perna e chutar a sacola .
2P: então , você vai fazer isso , vire e estenda a perna para fora e chute a bolsa .
PT: Então, você vai fazer esse turno, e estender essa perna e chutar a bolsa.
2P_TOK: então , você vai fazer isso , vire e estenda a perna para fora e **chute** a bolsa .

	Head 0	Head 1	Head 2	Head 3	Head 4	Head 5	Head 6	Head 7	Head 8	Head 9	Head 10	Head 11	Head 12	Head 13	Head 14	Head 15
squatting	0.07	0.13	0.05	0.05	0.05	0.01	0.24	0.07	0.09	0.01	0.01	0.05	0.01	0.09	0.07	0.22
kicking	0.15	0.09	0.33	0.14	0.08	0.41	0.01	0.10	0.28	0.77	0.07	0.03	0.01	0.02	0.14	0.06
raising	0.00	0.04	0.24	0.10	0.21	0.02	0.02	0.01	0.02	0.02	0.48	0.01	0.18	0.39	0.07	0.00
kneeling	0.51	0.32	0.02	0.06	0.04	0.10	0.62	0.59	0.32	0.01	0.02	0.55	0.00	0.02	0.26	0.50
bending	0.01	0.10	0.05	0.08	0.01	0.05	0.03	0.04	0.05	0.02	0.07	0.26	0.01	0.06	0.04	0.10
bowing	0.00	0.11	0.04	0.26	0.04	0.26	0.02	0.08	0.04	0.03	0.12	0.02	0.01	0.03	0.18	0.09
lifting	0.03	0.02	0.11	0.06	0.02	0.01	0.01	0.02	0.05	0.12	0.10	0.02	0.06	0.13	0.06	0.01
instructing	0.00	0.11	0.02	0.12	0.24	0.12	0.04	0.01	0.02	0.01	0.08	0.01	0.01	0.08	0.03	0.00
stretching	0.03	0.03	0.04	0.05	0.02	0.02	0.01	0.06	0.03	0.01	0.04	0.02	0.04	0.06	0.02	0.01
exercising	0.18	0.04	0.12	0.06	0.28	0.01	0.01	0.04	0.12	0.02	0.00	0.04	0.66	0.13	0.12	0.01

Figure 5.5: Multi-head attention of the second pass decoder of `Delib-AVF-THACE-A` to embeddings of the top 10 `THACE` categories detected from the same video segment, when generating the word “chute” (kick)

For videos that feature more narration than action, the 10 detected categories enjoy more or less similar attention on average. Figure 5.6 shows such an example, where the English sentence “So I would advise putting a lot of your practice time into the sand wedge” has no action verbs, and as a result the average-head attention has much more evenly spread focus on the action categories when generating the translated sentence.



Top10 SoftmaxEmbed, 6th layer
Head avg, Normal
EN: So, I would advise putting a lot of your practice time into the sand wedge.
Tr: so i would advise putting a lot of your practice time into the sand wedge
1P: então , eu aconselho colocar muito tempo de sua prática na cunha de areia .
2P: então eu aaaasou colocar muito tempo de prática na cunha de areia .
PT: Então, eu aconselho colocar um pouco do seu tempo de prática na cunha de areia.

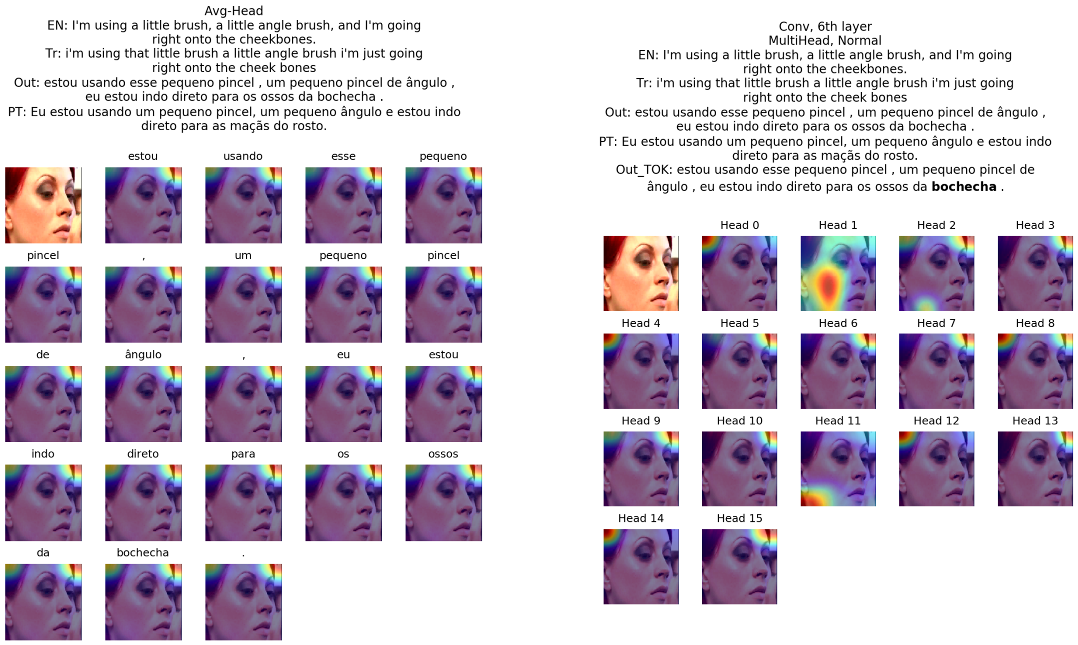
	então	eu	a@@	a@@	a@@	a@@	sou	colocar	muito	tempo	de	prática	na	cunha	de	areia	.
coaching	0.10	0.09	0.09	0.09	0.09	0.09	0.10	0.10	0.09	0.09	0.07	0.11	0.09	0.15	0.08	0.14	0.10
hitting	0.09	0.10	0.11	0.10	0.10	0.10	0.10	0.10	0.09	0.09	0.10	0.09	0.10	0.09	0.10	0.09	0.09
punting	0.13	0.11	0.09	0.09	0.09	0.10	0.10	0.11	0.12	0.12	0.11	0.15	0.12	0.13	0.12	0.14	0.16
gripping	0.11	0.10	0.10	0.09	0.09	0.09	0.09	0.10	0.11	0.11	0.12	0.10	0.11	0.09	0.11	0.10	0.10
putting	0.08	0.09	0.11	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.10	0.08	0.09	0.08	0.09	0.08	0.07
swinging	0.08	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.08	0.09	0.09	0.10	0.09	0.09
playing+sports	0.11	0.11	0.10	0.10	0.10	0.10	0.10	0.10	0.11	0.11	0.10	0.10	0.10	0.11	0.10	0.10	0.11
placing	0.10	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.10	0.10	0.12	0.09	0.11	0.08	0.11	0.08	0.09
instructing	0.08	0.09	0.10	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.09	0.07
blocking	0.12	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.11	0.10	0.11	0.10	0.11	0.09	0.11	0.09	0.11

Figure 5.6: 1^{st} and 15^{th} frames of a video segment & Average-head attention of the second pass decoder of `Delib-AVF-THACE-C` to embeddings of the top 10 `THACE` categories detected from the same video segment. EN: original subtitle; Tr: transcript sentence; 1P: first-pass decoding output; 2P: second-pass decoding output; PT: Portuguese reference; `2P_TOK`: BPE-ed & tokenised form of 2P

Visual Attention: Convolutional Layer Output

`CLO` features, coming from the convolutional layers, are usually assumed to contain more information than higher-level semantic information such as `THACE`. Indeed, we have found `CLO`-based transformer and deliberation networks to manifest their capability of capturing objects despite `CLO` being from an action recognition network. In fact, our observation is that it is easier to find examples where a `CLO`-based model focuses on an object in a video when the corresponding word is being translated, than to find instances where it exploits video regions for translating verbs. However, an important caveat is the aforementioned "good" localisation behaviour tends to exist only in a few heads, while the attention of the other heads is much less interpretable and arguably nonsensical, causing the average-head attention to appear useless. For each example in this section, we superimpose the upsampled attention heatmap onto the 8^{th} of the 16 equi-distant frames of a video which were fed to the action recognition network that generated the visual features.

Figure 5.7 shows a typical example of the description above from `Trans-AVF-CLO`. Inspecting only Figure 5.7a, one will easily reach the conclusion that the average attention here has no obvious way to be explained, since the attention for translating every word of the output sentence is focused on the upper right corner of the video. However, the multihead attention shown in Figure 5.7b when the word being generated is "bochecha" (cheeck) is not so simple — there are heads attending to the



(a) Average-head attention

(b) Multihead attention for "bochecha" (cheek)

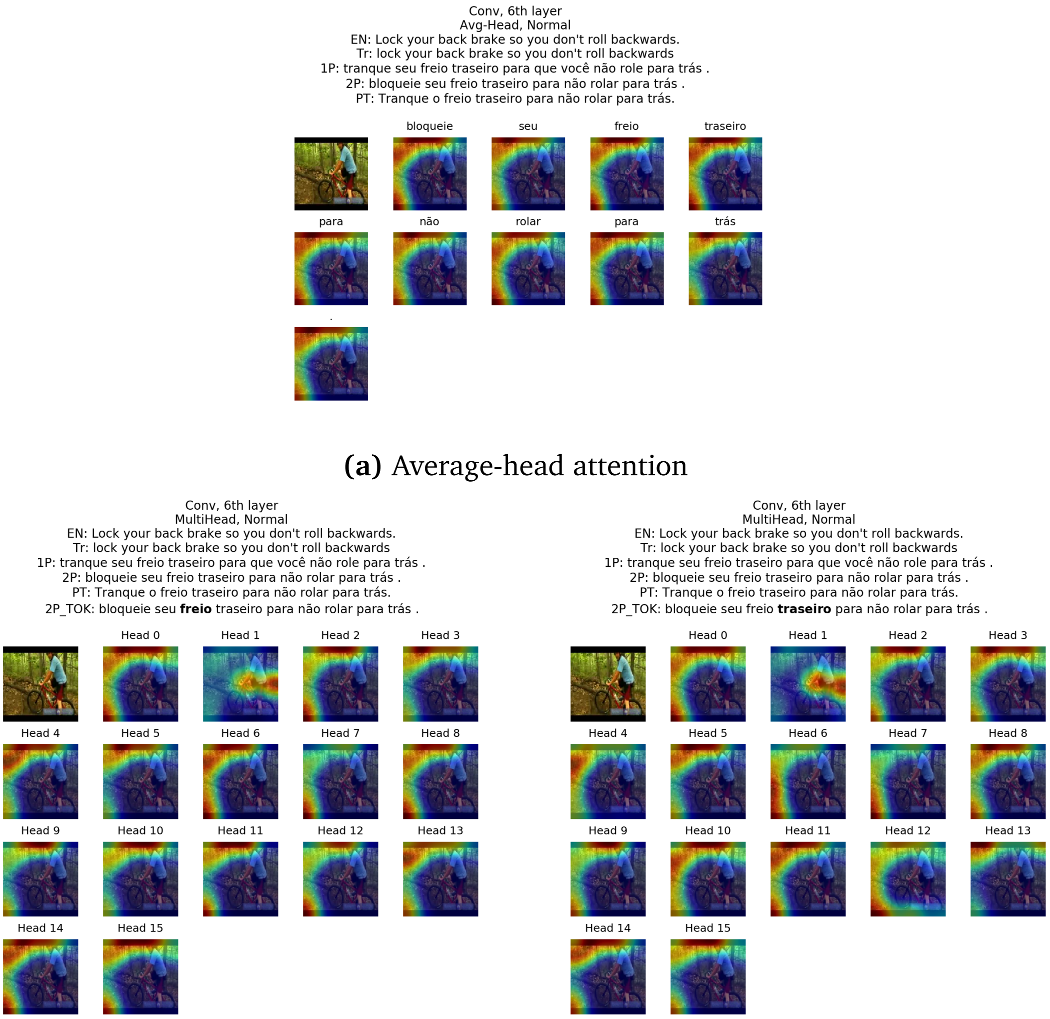
Figure 5.7: Average-head attention for whole sentence & Multihead attention for "bochecha" (cheek), from `Trans-AVF-CLO`. EN: original subtitle; Tr: transcript sentence; Out: decoding output; PT: Portuguese reference

upper left (e.g. Head 0) and lower left (e.g. Head 11) corners as well. In particular, the most heated attention area of Head 1 perfectly covers the cheek of the woman in the video. This sole head that has learned to attend to the correct region, however, is overwhelmed by the other heads with "wrong" attention when average-head attention is taken, leading to the upper-right-corner attention for "bochecha" shown in Figure 5.7a.

Apart from corners, edges are also commonly found to be where the average-head attention resides. In Figure 5.8a, the regional attention for every word in the output is again very similar and on the top and left edges predominantly without involving any objects. However, Head 1 has learned to focus on where the hands of the person meet the bicycle handle, when the model is translating "back brake" into "freio traseiro". Similarly, this one head is not able to move the average attention to the "right" place.

There are also examples where a head associates a region with an action instead of an object. In Figure 5.9b, the attention for the word "bater" (hit) is around the area covering the man's hand, arm and elbow, especially the latter. Then, again, the average head attention shifts to the lower left corner as depicted in Figure 5.9a. Generally, as mentioned before, it is harder to find individual head attention corresponding to actions as opposed to objects.

There are also a large number of examples where none of the heads attend to the

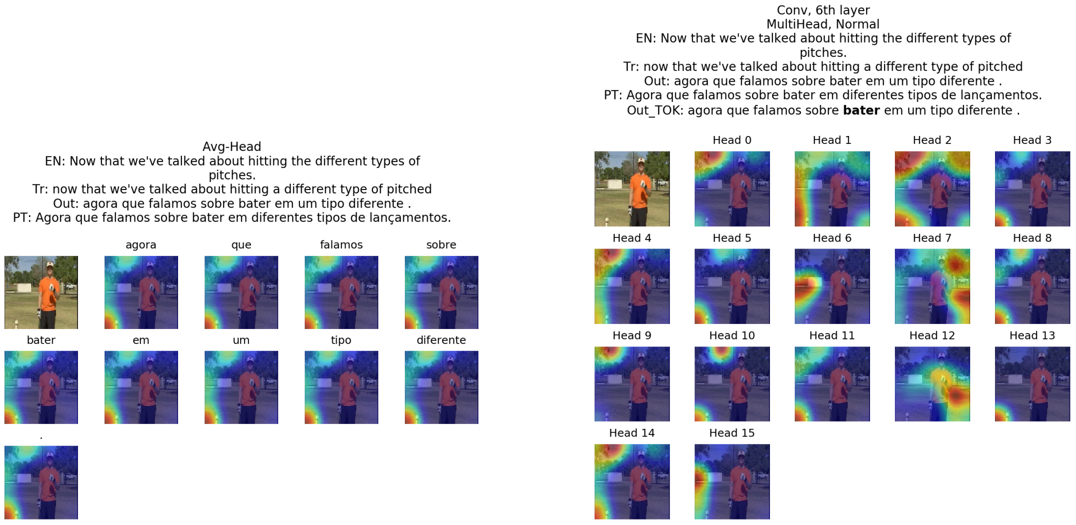


(a) Average-head attention

(b) Multihead attention for "freio" (brake) **(c)** Multihead attention for "traseiro" (rear)

Figure 5.8: Average-head attention for whole sentence & Multihead attention for "freio" (brake) and "traseiro" (rear), from `Delib-AVF-CLO-A`. EN: original subtitle; Tr: transcript sentence; 1P: first-pass decoding output; 2P: second-pass decoding output; PT: Portuguese reference; `2P_TOK`: BPE-ed & tokenised form of 2P

"interesting" region(s) of a video when translating an action or object word, especially where the source sentence is not closely associated with the video contents, e.g. when it is more narration than action. Nonetheless, the observation still holds that head diversity enables certain heads to attend to key regions in some scenarios. As to how this focusing-on-interesting-areas head behaviour can be encouraged and advanced, we leave it to future work.



(a) Average-head attention

(b) Multihead attention for "bater" (hit)

Figure 5.9: Average-head attention for whole sentence & Multihead attention for "bater" (hit), from `Trans-AVF-CLO`. EN: original subtitle; Tr: transcript sentence; Out: decoding output; PT: Portuguese reference

Chapter 6

Conclusions & Future Work

6.1 Contributions

In this thesis project, we primarily investigated the impact of visual features on multimodal machine translation based on How2 in the presence of corrupted source sentences. Our major contributions are as follows:

- We explored using multimodal transformers to translate How2 video subtitles in English with different degrees of verb masking into Portuguese, and achieved competitive performance. The BLEU results showed that our multimodal models were able to exploit the visual information to boost the translation performance. In particular, we found that the visual features from 3D ResNet-50, an action recognition network, were most helpful in terms of improving the model performance in cases of heavy verb masking, thus showing the importance of action-related visual information in helping the model recover from verb-related source corruption.

- We carried out incongruence analysis and human analysis for the multimodal transformers, offering more insight into the performance of those models. With incongruence, we recorded almost universal performance degradation, which was especially the case for the settings with verb masking, thus further demonstrating the effective utilisation of visual information by the models. Our human analysis on the other hand yielded concrete examples where the multimodal models recovered masked verbs using visual context, while at the same time revealed a discrepancy between human judgement and BLEU scores.

- We employed multimodal transformers and two types of multimodal deliberation networks for translating from English transcripts into Portuguese, a novel attempt at multimodal speech translation. Contrary to verb masking which is artificial and deterministic noise on the source side, transcribing introduces random corruption of the source. Also, multimodal cascade deliberation as an MMT architectural choice has never been tried before in other research to the best of our knowledge at the time of writing, so we used it for our experiments and found it to deliver similar performance as additive deliberation does.

- We devised the novel way of measuring congruent and incongruent multimodal speech translation results based on the faithfulness of the source transcript sentence with respect to its subtitle counterpart. By doing so, we discovered that the multimodel models suffered the most performance degradation on the examples that were most similar (i.e. faithful) to the subtitles instead of on the unfaithful examples where the models had been expected to exploit visual information to bridge the semantic gap between the transcripts and subtitles. In fact, we found incongruence to boost performance on those highly unfaithful examples in a number of cases. This, against the backdrop of our Phase 1 results, indicates that random noise is indeed harder from which to be recovered than artificially introduced source corruption, and that there is much work to be done on this. We also found that sensitivity to incongruence does not mean the visual modality necessarily boosts translation quality.

- We visualised the multihead attention of `THACE`- and `CLO`-based tranformers and deliebration networks to achieve more interpretability, a novel analysis that has not been attempted in other research to the best of our knowledge at the time of writing. As a result, we were able to see that, when the source sentence was closely related to the actions present in the video, the `THACE`-based models were able to narrow their attention down to a few prominent action categories and execute their decoding based on that. This focus of attention was noticeably less strong when the video was semantically distant from the sentence. `CLO`-based models, however, generally had misplaced attention to video regions on average, but some heads were found to be able to focus on the relevant parts of a video during the generation of certain words, in particular nouns as opposed to action verbs, despite the fact that `CLO` were from an action recognition network. Those few heads nonetheless were not powerful enough to shift the average attention to the "interesting" regions.

We also spent a considerable amount of time porting the deliberation network implementation from the old, `tensor2tensor` -1.3.0-based version to the latest `tensor2tensor` , and this is still ongoing work. Our plan is to carry out follow-up deliberation-based projects on the new version and then make it publicly available on GitHub.

Phase 1 of this project was undertaken as our participation in the How2 Challenge 2019, workshop of International Conference on Machine Learning. Our paper, *Predicting Actions to Help Predict Translations* (available at `https://arxiv.org/abs/1908.01665`), was accepted to the workshop, and our system "Attention over Image Features" is at the time of writing at the top of the Machine Translation Leaderboard (`https://srvk.github.io/how2-challenge/`) of the challenge.

6.2 Future Work

Future work of this project is chiefly about delving into the unanswered questions raised in previous chapters. Possible directions include:

- Investigating the discrepancy between human judgement and BLEU scores that arose in Phase 1. This can grow into a genuinely interesting and helpful quality estimation (Blatz et al., 2004; Specia et al., 2009) question for future MMT research.

- Probing further the link between model performance and sensitivity to incongruence. we discovered that these two are not correlated, therefore much insight can be offered into the still relatively black-box MMT models if the reason why and the way how multimodal models with already weak performance loses translation quality greatly in the presence of incongruence.

- Focusing on MMT quality improvement on unfaithful examples. In Phase 2, we had the surprising finding from our incongruence analysis that visual information did not help or even harmed multimodal performance on unfaithful transcripts. Since this is the subset where multimodality should enable substantial compensation for source corruption, it will be very helpful for source-corruption related MMT resarch if the reason behind this abnormality is identified.

- Ameliorating multihead attention of CLO-based models. An important observation about CLO-based transformer and deliberation models is that the few good heads that had learned appropriate attention were sidelined by the low-quality majority of heads. Previous work mentioned in Chapter 2.3 was able to achieve much more focused attention over image regions, therefore it is reasonable to believe transformer-based models should be capable of at least the same. Remedying or improving CLO-based multihead attention will be significant for transformer-based MMT models.

Bibliography

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. pages 1, 9

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D., and Frank, S. (2018a). Findings of the third shared task on multimodal machine translation. In *THIRD CONFERENCE ON MACHINE TRANSLATION (WMT18)*, volume 2, pages 308–327. pages 2, 13, 14

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D., and Frank, S. (2018b). Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323. Association for Computational Linguistics. pages 42

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321. pages 62

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. Association for Computational Linguistics. pages 43

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Soulié, F. F. and Hérault, J., editors, *Neurocomputing*, pages 227–236, Berlin, Heidelberg. Springer Berlin Heidelberg. pages 6

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Mercer, R., and Roossin, P. (1988). A statistical approach to language translation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*, COLING '88, pages 71–76, Stroudsburg, PA, USA. Association for Computational Linguistics. pages 1, 4

Caglayan, O., Aransa, W., Bardet, A., García-Martínez, M., Bougares, F., Barrault, L., Masana, M., Herranz, L., and Van de Weijer, J. (2017). Lium-cvc submissions for wmt17 multimodal translation task. *arXiv preprint arXiv:1707.04481*. pages 16, 19

Caglayan, O., Madhyastha, P., Specia, L., and Barrault, L. (2019a). Probing the need for visual context in multimodal machine translation. *CoRR*, abs/1903.08678. pages 2, 41

Caglayan, O., Sanabria, R., Palaskar, S., Barraul, L., and Metze, F. (2019b). Multimodal grounding for sequence-to-sequence speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8648–8652. IEEE. pages 23

Calixto, I., Liu, Q., and Campbell, N. (2017a). Doubly-attentive decoder for multimodal neural machine translation. *arXiv preprint arXiv:1702.01287*. pages 19, 20

Calixto, I., Liu, Q., and Campbell, N. (2017b). Incorporating global visual features into attention-based neural machine translation. *arXiv preprint arXiv:1701.06521*. pages 17

Calixto, I., Rios, M., and Aziz, W. (2019). Latent variable model for multi-modal translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6392–6405. pages 18

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. F. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64. Association for Computational Linguistics. pages 43

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (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. IEEE. pages 23

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*. pages 7

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (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*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics. pages 40

Delbrouck, J.-B. and Dupont, S. (2017). An empirical study on the effectiveness of images in multimodal neural machine translation. *arXiv preprint arXiv:1707.00995*. pages 19

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. pages 24

Elliott, D. (2018). Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978. pages 2

Elliott, D., Frank, S., Barrault, L., Bougares, F., and Specia, L. (2017a). Findings of the second shared task on multimodal machine translation and multilingual image description. *arXiv preprint arXiv:1710.07177*. pages 13

Elliott, D., Frank, S., Barrault, L., Bougares, F., and Specia, L. (2017b). 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*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics. pages 42

Elliott, D., Frank, S., and Hasler, E. (2015). Multilingual image description with neural sequence models. *arXiv preprint arXiv:1510.04709*. pages 1, 4, 15, 16

Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*. pages 2, 14

Elliott, D. and Kádár, A. (2017). Imagination improves multimodal translation. *arXiv preprint arXiv:1705.04350*. pages 17, 18

Grönroos, S.-A., Huet, B., Kurimo, M., Laaksonen, J., Merialdo, B., Pham, P., Sjöberg, M., Sulubacak, U., Tiedemann, J., Troncy, R., et al. (2018). The memad submission to the wmt18 multimodal translation task. *arXiv preprint arXiv:1808.10802*. pages 21

Hara, K., Kataoka, H., and Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555. pages 24

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., et al. (2018). Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*. pages 12, 13

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969. pages 21

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. pages 2, 16, 24

Helcl, J., Libovickỳ, J., and Variš, D. (2018). Cuni system for the wmt18 multimodal translation task. *arXiv preprint arXiv:1811.04697*. pages 14, 18, 20, 21, 25

Hitschler, J., Schamoni, S., and Riezler, S. (2016). Multimodal pivots for image caption translation. *arXiv preprint arXiv:1601.03916*. pages 1, 4

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. pages 7

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer. pages 21

Huang, P.-Y., Liu, F., Shiang, S.-R., Oh, J., and Dyer, C. (2016). Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 639–645. pages 21

Ismail, A., Wood, T., and Corrada Bravo, H. (2018). Improving long-horizon forecasts with expectation-biased lstm networks. pages 8

Ive, J., Madhyastha, Swaroop, P., and Specia, L. (2019). Distilling Translations with Visual Awareness. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pages 22, 25, 27, 32, 34, 37, 38, 47

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. pages 1, 4, 5

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. pages 34

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*. pages 22

Lala, C. and Specia, L. (2018). Multimodal lexical translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. pages 15

Lavie, A. and Agarwal, A. (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*, pages 228–231. Association for Computational Linguistics. pages 14

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436. pages 5

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551. pages 2, 15

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707. pages 38

Libovickỳ, J. and Helcl, J. (2017). Attention strategies for multi-source sequence-to-sequence learning. *arXiv preprint arXiv:1704.06567*. pages 20, 22

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer. pages 14, 21

Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297. pages 22

Madhyastha, P. S., Wang, J., and Specia, L. (2017). Sheffield multimt: Using object posterior predictions for multimodal machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 470–476. pages 16

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. pages 1, 5

Merity, S. (2016). Peeking into the neural network architecture used for google's neural machine translation. `https://smerity.com/articles/2016/google_nmt_arch.html`. Accessed: 2019-08-19. pages 6

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, Scottsdale, AZ, USA. pages 25

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41. pages 15

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfruend, D., Vondrick, C., et al. (2019). Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–8. pages 24, 31

Olah, C. (2015a). Neural networks, types, and functional programming. `http://colah.github.io/posts/2015-09-NN-Types-FP/`. Accessed: 2019-08-19. pages 7

Olah, C. (2015b). Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 2019-08-19. pages 8

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics. pages 14, 40

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543. pages 22

Popel, M. and Bojar, O. (2018). Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70. pages 33

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99. pages 21

Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., and Schiele, B. (2016). Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer. pages 22

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1. pages 5, 10

Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., and Metze, F. (2018). How2: A large-scale dataset for multimodal language understanding. *CoRR*, abs/1811.00347. pages 2, 24, 31

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681. pages 7

See, A. (2019). Lecture 8: Machine translation, sequence-to-sequence and attention. `http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture08-nmt.pdf`. Accessed: 2019-08-19. pages 9

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*. pages 33

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. pages 2, 15, 17, 21

Specia, L., Frank, S., Sima'an, K., and Elliott, D. (2016). A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 543–553. pages 13

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37. pages 62

Specia, L., Wang, J., Ostapenko, A., Lee, J., and Madhyastha, P. S. (2019). Read, spot and translate (under review). pages 21, 22

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112. pages 5

Tilk, O. and Alumäe, T. (2016). Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*. pages 35

Tutschku, K. (1995). Recurrent multilayer perceptrons for identification and control: The road to applications. *Univ. Würzburg, Germany, ser. Research Report Series*. pages 7

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., et al. (2018). Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*. pages 32

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008. pages 1, 10, 11, 32

Weaver, W. (1955). Translation. *Machine translation of languages*, 14:15–23. pages 1, 4

Werbos, P. J. et al. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560. pages 5, 7

Xia, Y., Tian, F., Wu, L., Lin, J., Qin, T., Yu, N., and Liu, T.-Y. (2017). Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794. pages 3, 12, 13, 27, 32

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500. pages 24

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. pages 18

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (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. pages 14

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2019). Dive into deep learning. chapter 8.10. `http://www.d2l.ai`. pages 6

Appendices

Appendix A

Ethics Checklist

	Yes/No
Section 1: HUMAN EMBRYOS/FOETUSES	
Does your project involve Human Embryonic Stem Cells?	No
Does your project involve the use of human embryos?	No
Does your project involve the use of human foetal tissues / cells?	No
Section 2: HUMANS	
Does your project involve human participants?	Yes
Section 3: HUMAN CELLS / TISSUES	
Does your project involve human cells or tissues? (Other than from Human Embryos/Foetuses i.e. Section 1)?	No
Section 4: PROTECTION OF PERSONAL DATA	
Does your project involve personal data collection and/or processing?	No
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	No
Does it involve processing of genetic information?	No
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.	No
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?	No
Section 5: ANIMALS	
Does your project involve animals?	No
Section 6: DEVELOPING COUNTRIES	
Does your project involve developing countries?	No
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?	No

Could the situation in the country put the individuals taking part in the project at risk?	No
Section 7: ENVIRONMENTAL PROTECTION AND SAFETY	
Does your project involve the use of elements that may cause harm to the environment, animals or plants?	No
Does your project deal with endangered fauna and/or flora /protected areas?	No
Does your project involve the use of elements that may cause harm to humans, including project staff?	No
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?	No
Section 8: DUAL USE	
Does your project have the potential for military applications?	No
Does your project have an exclusive civilian application focus?	No
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?	No
Does your project affect current standards in military ethics e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?	No
Section 9: MISUSE	
Does your project have the potential for malevolent/criminal/terrorist abuse?	No
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?	No
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?	No
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?	No
SECTION 10: LEGAL ISSUES	
Will your project use or produce software for which there are copyright licensing implications?	No
Will your project use or produce goods or information for which there are data protection, or other legal implications?	No
SECTION 11: OTHER ETHICS ISSUES	
Are there any other ethics issues that should be taken into consideration?	No

Table A.1: Ethics Checklist

Summary

We had human participants who assessed our systems in Phase 1 of the project, as explained in Chapter 5.1.3. The four people chosen were my supervisor, Prof. Lucia Specia, and three current and former colleagues of hers. They are all adults, and were selected based on the fact that they are all native Portuguese speakers proficient in English and well-versed in natural language processing. We sent an email to each of them detailing our need for human analysis and asking them if they could help, and they consented to participation with their affirmative replies. They agreed to take part in the analysis without being paid, and we thank them for their valuable help.

Hence, there are no legal or professional issues involved in this project.