Improving Neural Machine Translation with External Information Thesis Proposal

Jindřich Helcl
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
helcl@ufal.mff.cuni.cz

Abstract

Results of previous studies in the field suggest that neural machine translation (NMT) models can achieve better performance by exploiting information from external sources. In this thesis proposal, we categorize a variety of types of the enhanced models into four classes – multimodal, multilingual, linguistically co-supervised, and linguistically inspired. We summarize some of the approaches in each of these categories found in the literature. A part of this work is a development of a sequence-to-sequence learning toolkit designed for fast prototyping of various kinds of experiments. We report results of experiments proposed by us in the past, which were mainly based on the multimodal models. In the future, we plan to focus on the linguistically co-supervised models, which could make use of the abundance of linguistic annotation available.

1 Introduction

With the advent of deep learning, the model architectures we use for natural language processing (NLP) tasks have simplified to such level that they do not use any external information, be it an expert knowledge (e.g. linguistic annotations) or an equivalent information expressed in a different modality (e.g. photos or sound). This means that we are losing valuable sources of linguistic information which according to many studies retain their value in a number of NLP tasks, such as machine translation (Sennrich and Haddow, 2016; Eriguchi et al., 2017), question answering (Zhang et al., 2017), or sentiment analysis (Matsumoto et al., 2005).

In this text, we categorize the models that exploit the external information to *multimodal*, *multi-source*, *linguistically co-supervised*, and *linguistically inspired*. We present a selected number of works from each of those categories.

We present our contributions to the category of multimodal models. We show the results of the experiments conducted in the previous years, mainly as submissions to the shared task at Conference on Machine Translation (WMT).

For our future research, we plan to contribute to the category of linguistically co-supervised models applied to neural machine translation (NMT). The main feature of linguistically co-supervised models is the use of linguistic annotations in the training process, rather than imposing hard constraints to the network architecture itself. This can be achieved for example in the multi-task learning scenario.

We see a particular particularly promising research direction in using our in-house annotated corpora, such as Prague Czech-English Dependency Treebank (Hajič et al., 2012), that may show the usefulness of these high-value resources.

The text of this proposal is structured as follows. In Section 2, we describe the standard models used for NMT. Section 3 gives an overview of existing methods of incorporating linguistic information to deep learning models. Section 4 sums up the experiments already conducted and presents their results. In Section 5, we present a number of ideas for the following experiments. We conclude in Section 6.

2 Neural Machine Translation

The goal of NMT is to train a neural network to read a sentence in the source language and generate a corresponding sentence in the target language. In a broader context, this task can be

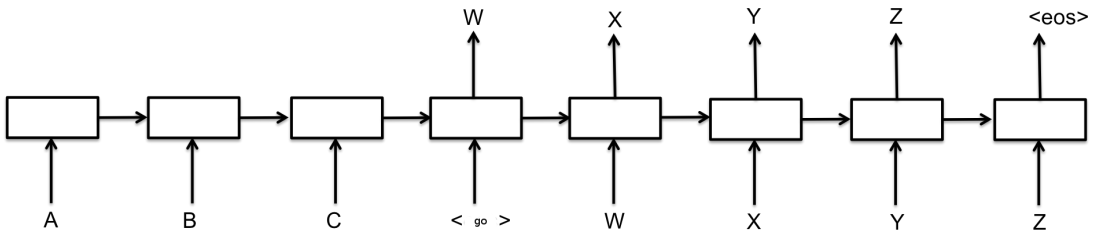


Figure 1: Scheme of the classic sequence-to-sequence model by Sutskever et al. (2014)

viewed as an instance of a sequence-to-sequence (S2S) learning problem (Sutskever et al., 2014).

In S2S learning, first, the source sequence $X = (x_1, \ldots, x_{T_x})$ is *encoded* using a recurrent neural network (RNN) in a fixed-length vector h_{T_x}. Next, in each step, the output of the previous step is fed to the *decoding* RNN, which updates its hidden state s_{i-1} and outputs a new state s_i and a new output symbol y_i. The step procedure is repeated until a special symbol <eos> is emitted.

Since plain RNNs can suffer from the vanishing gradients problem, they were replaced in the first NMT experiments with Long Short-Term Memory networks (LSTM; Hochreiter and Schmidhuber, 1997). Another variant of the RNN which was used in our experiment is the Gated Recurrent Unit network (GRU, Cho et al., 2014). Both of these variants introduce a gating system which can handle vanishing gradients problem in long-range dependencies through additive memory updates with linear derivatives.

The following equations describe a GRU-based sequence-to-sequence model:

$$h_j = \mathrm{GRU}_{enc}(h_{j-1}; E_{enc}\, x_j), \quad (1)$$

$$s_0 = \tanh(V_s h_{T_x} + b_s), \quad (2)$$

$$s_i = \mathrm{GRU}_{dec}(s_{i-1}; E_{dec}\, y_{i-1}) \quad (3)$$

where $h_0, \ldots, h_{T_x}$ are hidden states of the encoder (h_0 being a null-vector), E_{enc} and E_{dec} are the so-called *embedding* matrices, which assign a real-valued vector to each word in the vocabulary, V_s and b_s are additional trainable parameters of the linear projection of h_{T_x}, and $s_0, \ldots, s_{T_y}$ are hidden states of the decoder. As y_0, we use a special <go> symbol that denotes the start of the sentence. T_x and T_y are the lengths of the source and target sequence respectively. We use semicolon to denote vector concatenation.

The actual output word is selected based on the output of the decoder network:

$$t_i = W_o(s_i; E_{dec}\, y_{i-1}) + b_o, \quad (4)$$

$$P(\hat{y}_i | x, \mathbf{y}_{<i}) \propto \exp t_i, \quad (5)$$

$$\hat{y}_i = \operatorname*{argmax}_y P(y | x, \mathbf{y}_{<i}). \quad (6)$$

One of the most significant improvements over the S2S baseline model is introduction of the attention mechanism (Bahdanau et al., 2014). Similarly to content-based addressing in Neural Turing Machines (Graves et al., 2014), the attention mechanism allows us to use context-sensitive information in the decoding phase.

In each decoder step i, the output of the attention mechanism is a context vector c_i, which is then concatenated to the input of the decoder RNN. The Equations 3 and 4 are thus adjusted to work with the context vector as follows:

$$s_i = \mathrm{GRU}_{dec}(s_{i-1}; c_i; E_{dec} y_{i-1}). \quad (7)$$

$$t_i = W_o(s_i; c_i; E_{dec}\, y_{i-1}) + b_o. \quad (8)$$

In the i-th step of the decoder, the context vector c_i is defined as a weighted sum of the encoder states using the attention distribution α_{ij}, which is in turn obtained by normalizing the attention energies e_{ij}:

$$e_{ij} = v_a^\top \tanh(W_a s_i + U_a h_j), \quad (9)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad (10)$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \quad (11)$$

The trainable parameters W_a and U_a are projection matrices that transform the decoder and encoder states s_i and h_j into a common vector space and v_a is a weight vector over the dimensions of this space. For the sake of clarity, bias terms (applied every time a vector is linearly projected using a weight matrix) are omitted.

En:	A *wall* divided the city.	
De 1:	Eine *Wand* teilte die Stadt.	×
De 2:	Eine *Mauer* teilte die Stadt.	✓

Figure 2: An illustration of disambiguation using the visual features. Example taken from Specia et al. (2016).

The model is trained by minimizing a loss function defined over the output probability distribution. A commonly-used loss function for NMT models is the negative log-likelihood:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P(y_i|x_i, \mathbf{y}_{<i}, \theta) \qquad (12)$$

where N is the number of training examples, and θ is the set of all trainable parameters.

3 Related Work

There is a significant amount of evidence that suggests using external information can help improve NMT.

In this section, we divide these studies into four categories according to the type of the external information: First, Section 3.1 deals with multimodal translation – exploiting information provided as an additional modality, specifically images. Second, Section 3.2 describes approaches based on using linguistic annotations. Third, Section 3.4 shows results of multilanguage translation models. Finally, Section 3.3 present studies that are linguistically inspired, although do not use the linguistic annotation data directly.

3.1 Multimodal Translation

Multimodal translation is a task of generating the target sentence given a source sentence and an additional information in a different modality. In this section, we will focus on translation models enhanced with images. This means our task is to translate an image caption from one language to another, provided we can access the image itself.

Elliott et al. (2015) argue that including the image features in the model architecture can help the translation system to disambiguate between different meanings. Figure 2 gives an example where visual features provide such information for disambiguation.

In their work, they employ a S2S model as described in Equations 1–6 which is provided with the visual features obtained from the penultimate layer of the VGG-16 object recognition network (Simonyan and Zisserman, 2014). In each step of the encoder and/or encoder, they include the visual feature vector among the inputs of the RNN (i.e. they concatenate the visual feature vector with the rest of the inputs in Equations 1 and 3). Although they argue the models benefit from the added modality, their best model did not outperform the textual baseline in terms of BLEU/METEOR (Papineni et al., 2002; Denkowski and Lavie, 2011).

Xu et al. (2015) introduced attention mechanism used for image captioning – a task of generating a textual description of an image. Instead of attending to a set of states of an RNN encoder, they employ the same technique over the components of the last convolutional layer of the VGG-16 network. This way, they were able to extract context-sensitive image features relevant for a given RNN decoder state.

In their WMT16 submission, Caglayan et al. (2016a) tried to combine the attention distribution instead of combining the resulting context vectors. They perform the weighted sum from Equation 11 over the states from both modalities:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij}^{txt} h_j + \sum_{j=1}^{196} \alpha_{ij}^{img} v_j \qquad (13)$$

where v_j is the j-th vector in the last convolutional layer (which contains 196 such vectors that correspond to the image down-scaled to 14-by-14 regions). In this case, they made a strong assumption that the network can be trained in such a way that the encoder states and the states of the convolutional network occupy the same vector space. Therefore, the score of their multimodal MT system remained far below the text-only setup.

Soon after, Caglayan et al. (2016b) introduced the multimodal attention mechanism. In their version, they compute the context vectors for both im-

age and the source sentence in each decoder step. They adapt Equations 7 and 8 by supplying either the sum or the concatenation of the single-modal context vectors as c_i. In this work, they reported slight improvements over the textual baseline on the Multi30k dataset, which was made available for the WMT16 multimodal translation task (Elliott et al., 2016). A significant difference between this work and the previous approaches is that instead of the VGG-19 network, they use the ResNet-50 network for extracting the visual features (He et al., 2016).

More recent state-of-the-art results on the Multi30k dataset were achieved by Calixto et al. (2017). The best-performing architecture uses the last fully-connected layer of VGG-19 network for the initialization of the decoder and attends only to the states of the RNN (text) encoder.

Elliott and Kádár (2017) brought further improvements by introducing the "imagination" component to the neural network architecture. Given the source sentence, the network is trained to output the target sentence jointly with predicting the image vector. The model uses the visual information only as a regularization and thus is able to use additional parallel data without accompanying images.

3.2 Linguistically Co-Supervised Models

Training of linguistically co-supervised models requires an external source of linguistic annotation. The linguistic annotation itself can be of any kind. In most of the published studies, we encounter the use of part-of-speech (POS) tags, syntax, or named entities.

Sennrich and Haddow (2016) introduce linguistic features (lemma, POS, and dependency label) as an additional input to the decoder, as previously applied to neural language models (Alexandrescu and Kirchhoff, 2006). In this scheme, each feature type (*factor*) has its own embedding matrix. The embedded factors are concatenated together with the word embedding (Equation 1). The rest of the architecture is identical to the standard NMT model described above. The results on the WMT16 test data show that using all factors together gives significant improvements over a strong baseline in terms of BLEU and chrF3 (Popović, 2015). The study also shows a contrastive comparison of the effect of different feature types to the translation quality.

Similarly, Li et al. (2017) focus on providing the source syntax information to the encoder. They propose three encoder models which operate on the input sentences combined with sequences of labels from linearized phrase-structure trees. First, *parallel encoder* is composed of two RNNs – one processes the sequence of words, and the other processes the the sequence of syntactic labels. The states of the word encoder are then concatenated with the syntax encoder states which correspond to the syntactic labels of the words (leaves of the tree). Second, *hierarchical encoder* is also divided into two RNNs. Unlike the parallel encoder, the hierarchical encoder first processes the parse tree and then concatenates the tree-leaf RNN states with embeddings of corresponding words. After that, the second RNN is run over the updated embeddings. Finally, *mixed encoder* is a single RNN that operates on a (mixed) sequence of both syntactic labels and words. This sequence is ordered so the words are inserted to the network immediately after their corresponding labels. The mixed encoder model is shown to have the best performance of all the proposed models. It outperforms the RNNSearch model of Bahdanau et al. (2014) on NIST MT02–MT05 datasets.

Bastings et al. (2017) use so-called graph-convolutional networks in encoders. They use a dependency parser to preprocess the training data so that each word has its dependency labels and a pointer to its head in the tree. The graph-convolutional encoder processes the input sequence in multiple layers. In each layer, the information from the dependent nodes is mixed with the state of the head. This way, the information from one node propagates through the dependency graph layer-by-layer.

The following approaches employ multi-task learning techniques. The loss function from Equation 12 can be changed to contain the joint probability of predictions for all the different types of labels.

Eriguchi et al. (2017) found that training the model to parse and translate helps NMT. They present a hybrid model between NMT and recurrent neural network grammar (RNNG, Dyer et al., 2016). The model replaces the parser's buffer with the decoder state, which enables the decoder to control the parser. For training the model, instead of using manually annotated parallel corpora, they use the SyntaxNet parser (Andor et al.,

2016) and generate the training parse trees automatically. This model gains significant improvements over NMT baseline on WMT newstest2016 on three of four language pairs that has been experimented with.

Another example of multi-task learning is the work of Tamchyna et al. (2017). Their NMT system works in two steps. First, they train the system to produce the lemma and POS tag of the target word (in a serialized form). Second, they use a deterministic generator to produce the target words from the lemmas and tags. In order to train the system, they preprocess the target side of the training data with morphological analyzers and add the lemma and POS information to the training corpus. With this method, they obtain significant improvements, especially in English to Czech translation, where the target language is morphologically rich.

Finally, Niehues and Cho (2017) systematically explore the multi-task learning setups for three tasks – machine translation, POS tagging, and named-entity recognition. They experiment with three different levels of sharing model parts. First, the *shared encoder* model, uses a single encoder for the source sentence and separate decoders and attention mechanisms for each of the tasks. Second, the *shared attention* model, shares not only the encoder, but also the attention mechanism across the tasks. This is achieved by sharing the U_a and v_a parameters from Equation 9. Third, the *shared decoder* model shares the encoder, the attention model, and the decoder for all the tasks. The first part of the model that is not shared among task is therefore the W_o and b_o parameters of Equation 8.

3.3 Linguistically Inspired Models

There are more ways of exploiting linguistic information than providing the training procedure directly with explicit annotations. The category of linguistically inspired models brings together methods that apply changes to the model architecture which are to some extent inspired by linguistic theory.

To tackle the problem of translating low-frequency words, Arthur et al. (2016) incorporate translation lexicons into NMT. The probabilities from the translation lexicon are first converted into predictive probabilities over next word using the attention distribution (Equation 10). They present two ways of combining the lexicon predictive probabilities with the probabilities outputted by the model. First, the *model bias* approach introduces a hyper-parameter that controls how much the model is biased towards relying on the lexicon. Second, the *linear interpolation* approach uses a trainable parameter which controls the interpolation of the lexicon-based probabilities and the output probabilities.

Another linguistically inspired approach is source-side latent graph parsing (Hashimoto and Tsuruoka, 2017). In these experiments, a pre-trained LSTM dependency parser is used to process the input, and the parse tree is provided as input of a recurrent decoder with attention mechanism. The parameters of the LSTM parser remain trainable during training of the translation model, without using any more annotated data.

3.4 Multi-source NMT

Multi-source NMT is a task of translating a sentence (which can be in one of multiple source languages) to a sentence in a common target language.

One of the first experiments with multi-source NMT was conducted by Zoph and Knight (2016). They use a trilingual parallel corpus to train a model that translates to English using a French and German source sentence. In their paper, however, they do not report scores of the model when one of the source sentences is missing.

Johnson et al. (2016) present an extension of the Google NMT model (Wu et al., 2016) that works with many source and target languages. They describe separately the scenarios for many-to-one, one-to-many, and many-to-many translation models. In these models, they include a special token to the end of the source sentence which specifies the target language (so the encoder can adjust to the information). Interestingly, they found that for a fixed language pair, adding data with a different source language can help the translation quality of the original language pair.

4 Experiments

This section presents experiments we conducted so far and discuss the results. In the first part, Section 4.1 describes our toolkit designed for fast prototyping of neural architectures. Section 4.2 gives an overview of our models submitted to the WMT 16 multimodal translation and post-

editing tasks. Section 4.3 presents our method for combining attention mechanisms in multi-source setup. Section 4.4 presents the revisited models used in our submission to the WMT 17 multimodal translation task.

4.1 Neural Monkey

We developed a toolkit for fast prototyping of neural network models for our experiments. This section gives a brief description of the toolkit and its main features.

Neural Monkey (Helcl and Libovický, 2017) is an open-source software implemented in Python3 using the TensorFlow library (Abadi et al., 2016). It's goal is to provide a higher level API which enables users to quickly prototype and train new architectures without requiring knowledge of the implementation details. For that reason we try to use as big abstract building blocks as possible.

Unlike other toolkits, like *tfLearn*[1] or *Lasagne*,[2] our building blocks are more abstract objects (e.g. encoders or classifiers) rather than individual network layers. These objects are parametrized so that their actual properties (e.g. dimensionality of embeddings or hidden states, dropout probability, number of layers) can be set from distant perspective. This design decision allows us to have the experiment configuration placed elsewhere than the actual code, in a separate comprehensive configuration file.

Neural Monkey is still under development and its mission is to become an ever-growing collection of recent innovations in S2S learning so its users are able to easily try out the models for their specific tasks and datasets. With its simple experiment management, it is used as a ready-made easily-extensible toolkit for experiments with machine translation, image captioning, text classification tasks or scene text recognition.

4.2 WMT 16 Multimodal Translation and Automatic Post-Editing Tasks

This section describes our submissions to the WMT 16 automatic post-editing and multimodal translation shared tasks (Libovický et al., 2016). The goal of the automatic post-editing task is to automatically correct a machine-generated translation while having access also to the corresponding source-language sentence. As mentioned above, the goal of multimodal translation is to translate an image description from one language to another, with access to the image itself.

Our method uses the neural translation model with attention by Bahdanau et al. (2014) and extends it to include an arbitrary number of encoders (see Figure 3). Each input sentence enters the system simultaneously in several representations $X^{(k)}$. An encoder used for the k-th representation is either a bidirectional GRU network (for textual input) or the VGG-16 convolutional neural network (CNN).

The initial state of the decoder is computed as a weighted combination of the final states of all the encoders. As the final state of a CNN encoder, we use the activation vector from the penultimate layer of the network.

The attention is computed over each encoder separately as described in Equations 9, 10, and 11. The context vectors are concatenated prior computing the distribution over the output vocabulary. Here is a revisited version of Equation 8 that reflects this approach:

$$t_i = W_o(s_i; E_{dec}y_{i-1}; c^{(1)}; \ldots; c^{(k)}) + b_o. \quad (14)$$

For the automatic post-editing shared task, we use two encoders. First encoder processes the source sentence and the second encoder processes the machine-generated translation. Rather than training the model to generate the post-edited sentence directly, we trained the model to output a sequence of edit operations. An edit operation could be either a word from the output vocabulary (in which case the word got inserted to the output), or one of two special symbols, *keep* or *delete*. In order to generate the final output, the edit operation sequence was applied to the machine-generated translation using a deterministic procedure.

For the multimodal translation task, we first translated the training dataset with Moses (Koehn et al., 2007). Then, we trained a model with three encoders – one CNN encoder for the image (with fixed parameters), and two textual encoders for the source image description and the Moses translation.

Since the target language for both tasks was German, we also did language dependent text preprocessing. Before training, we split the contracted prepositions and articles (*am* ↔ *an dem*, *zur* ↔ *zu der*, ...) and separated some pronouns from their case ending (*keinem* ↔ *kein -em*, *unserer* ↔ *unser -er*, ...). We also tried splitting

[1] https://github.com/tflearn/tflearn

[2] https://github.com/Lasagne/Lasagne

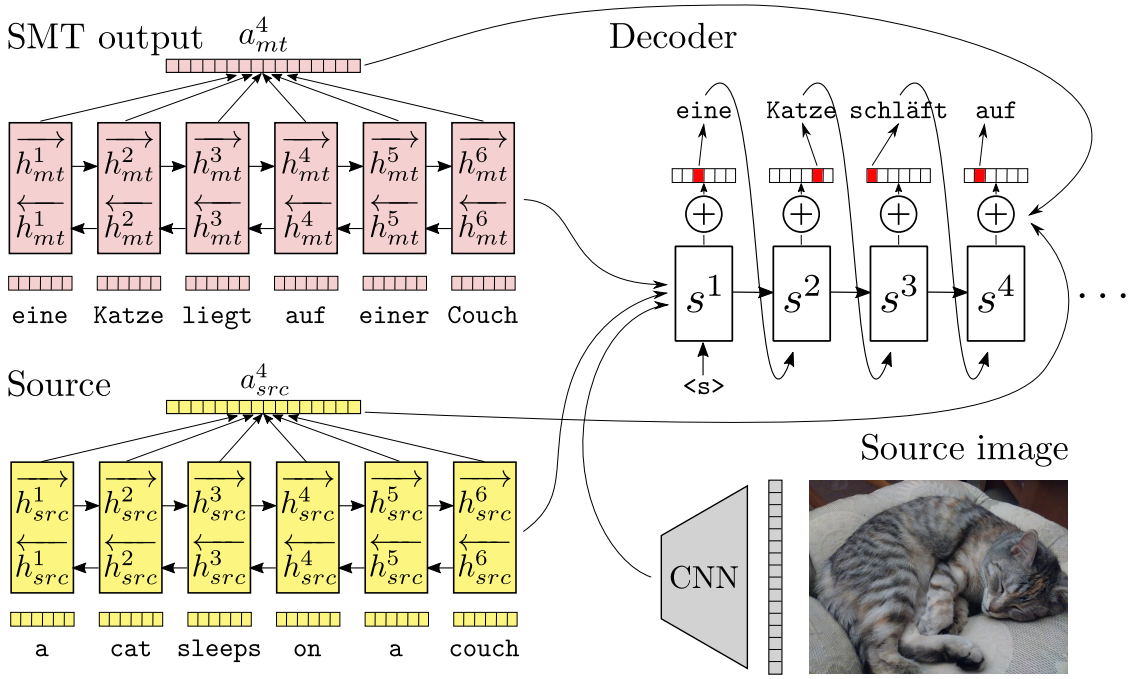


Figure 3: Multi-encoder architecture used for the multimodal translation.

compound nouns into smaller units, but on the relatively small data sets we have worked with, it did not bring any improvement.

In this submission, we also experimented with some improvements of S2S learning such as scheduled sampling (Bengio et al., 2015), noisy activation function (Gülçehre et al., 2016), or linguistic coverage model (Tu et al., 2016). However, none of these methods was able to improve the performance of our systems.

4.3 Attention Combination Strategies for Multi-Source S2S Learning

In this section, we describe *flat* and *hierarchical* attention combination strategies (Libovický and Helcl, 2017). We employ this architecture in S2S learning with multiple input sequences of various modalities and a single RNN decoder.

The prior approaches to multi-source S2S learning do not explicitly model different importance of the inputs to the decoder (Firat et al., 2016; Zoph and Knight, 2016). An example motivation scenario is multimodal translation, where we might expect the image description to be the primary source of information, whereas the image features would help mainly with visual disambiguation.

We describe the two combination strategies in more detail.

Flat attention combination In this strategy, we project the states of all encoders into a common vector space and then compute attention over the projected vectors.

The difference between the concatenation of the context vectors (as seen e.g. in Caglayan et al., 2016b) and the flat attention combination is that the α_i coefficients are computed jointly for all encoders (modified Equation 10):

$$\alpha_{ij}^{(k)} = \frac{\exp(e_{ij}^{(k)})}{\sum_{n=1}^{N}\sum_{m=1}^{T_x^{(n)}}\exp\left(e_{im}^{(n)}\right)} \qquad (15)$$

where $T_x^{(n)}$ is the length of the n-th encoder input sequence and $e_{ij}^{(k)}$ is the attention energy of the j-th state of the k-th encoder in the i-th decoding step. These attention energies are computed according to Equation 9 – the parameters v_a and W_a in the equation are common for all the encoders, whereas the matrix U_a is encoder-specific and serves as a projection matrix from each encoder state space into a single shared vector space.

Since the states of the encoders occupy different vector spaces, they can have different dimensionality. Hence, the context vector cannot be computed as their weighted sum (Equation 11). Therefore, we project the encoder states into a single

space using linear projections:

$$c_i = \sum_{k=1}^{N} \sum_{j=1}^{T_x^{(k)}} \alpha_{ij}^{(k)} U_c^{(k)} h_j^{(k)} \quad (16)$$

where $U_c^{(k)}$ are additional trainable parameters.

The projection matrices $U_a^{(k)}$ and $U_c^{(k)}$ project the states of one encoder into vector spaces with equal dimensionality. In our experiments, we tried both setting these projection matrices equal and training them separately.

Hierarchical attention combination This combination strategy divides the computation of the attention distribution into two steps: First, it computes the context vector for each encoder independently using Equations 9–11. Second, it projects the context vectors into a shared vector space (Equation 17), computes another distribution over the projected context vectors (Equation 18), and their corresponding weighted average (Equation 19):

$$e_i^{(k)} = v_b^\top \tanh(W_b s_i + U_b^{(k)} c_i^{(k)}), \quad (17)$$

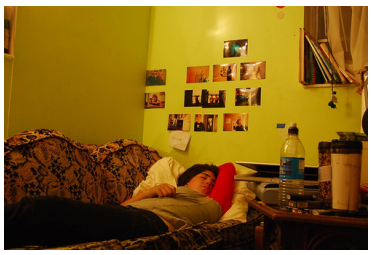
$$\beta_i^{(k)} = \frac{\exp(e_i^{(k)})}{\sum_{n=1}^{N} \exp(e_i^{(n)})}, \quad (18)$$

$$c_i = \sum_{k=1}^{N} \beta_i^{(k)} U_c^{(k)} c_i^{(k)} \quad (19)$$

where $c_i^{(k)}$ is the context vector of the k-th encoder, additional trainable parameters v_b and W_b are shared for all encoders, and $U_b^{(k)}$ and $U_c^{(k)}$ are encoder-specific projection matrices, that can be set either equal or trained independently, similarly to the case of flat attention combination.

From all the proposed methods, the best performing was the hierarchical attention combination with independently trained encoder projections (matrices U_a and U_c). In this setting, we were able to significantly outperform the concatenation baseline.

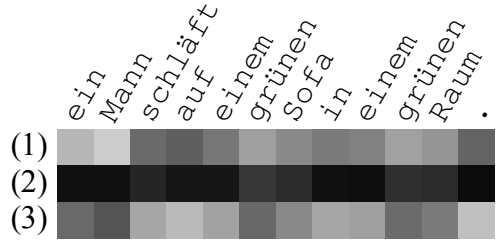
Both the hierarchical and the flat combination strategies provide an explicit way to interpret different importances of each inputs. In Figure 4, you can see the hierarchical attention distribution. The image also include the sentinel gate (Lu et al., 2016), which allows the decoder not to attend to any of the input encoders. Experimenting with the sentinel gate, however, did not bring any improvements and the details are out of the scope of this proposal.



Source: a man sleeping in a green room on a couch .

Reference: ein Mann schläft in einem grünen Raum auf einem Sofa .

Output with attention:



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

Figure 4: Visualization of hierarchical attention in MMT. Each column in the diagram corresponds to the weights of the encoders and sentinel. Note that the despite the overall low importance of the image encoder, it gets activated for the content words.

4.4 WMT 17 Multimodal Translation Task

Our submission to this year's WMT multimodal shared task (Helcl and Libovický, 2017) employs the hierarchical attention combination described above. Figure 5 shows the diagram of the model.

We expanded the training dataset using additional data acquired from several sources. First, we back- translated German descriptions that were part of the Multi30k dataset (Elliott et al., 2016). This gave us six times more training data because beside the German translations of the image descriptions, the Multi30k dataset also contains five independent German descriptions for each image. Second, using a language model trained on the image descriptions, we selected similar sentence pairs from the parallel SDEWAC corpus (Faaß and Eckart, 2013) and German parts of WMT parallel corpora, such as EU Bookshop (Skadiņš et al., 2014), News Commentary (Tiedemann, 2012), and CommonCrawl (Smith et al., 2013).

We also reported a number of negative results. First, we tried to rescore the top-k best hypotheses using a multimodal classifier. Second, we

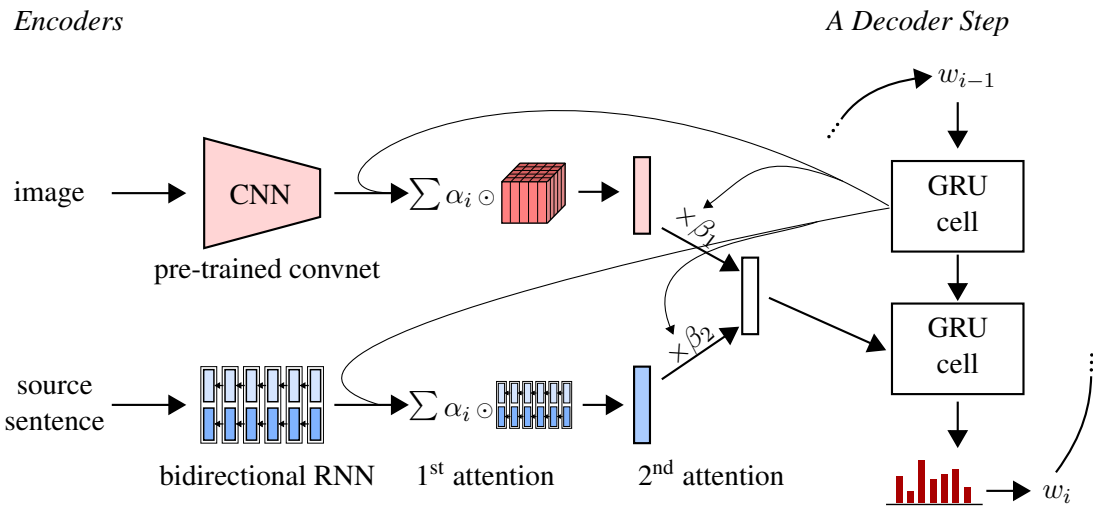


Figure 5: An overall picture of the multimodal model using hierarchical attention combination on the input. Here, α and β are normalized coefficients computed by the attention models, w_i is the i-th input to the decoder.

switch the optimization criterion from negative log-likelihood to optimize directly towards BLEU using self-critical training (Rennie et al., 2016). Despite the current negative result, we believe that these methods may be relevant for future development in the field.

5 Future Work

In the following paragraphs, we outline the proposal of the future work. The main topic of the research is to explore ways of using external information in order to improve neural machine translation.

Explore multi-task learning solutions. In our future work, we aim to explore more multi-task learning schemes, which will serve as a platform for models that learn to predict target-side linguistic features. Not only this has the potential to be beneficial for the primary task (i.e. translation), but it can also show which architectures are able to capture the linguistic information implicitly and which architectures need supervision from human annotation.

Specialize on Czech annotated data. We will focus our experiments on exploiting the abundance of high-quality annotated corpora available at our institute. As we see in the literature, for the English language, including the linguistic annotation to the model has a great potential of improving the performance. We believe that the scale of the improvement will be even higher for a language with rich morphology and non-projective dependency structures, such as Czech. Moreover, the Czech language is a perfect candidate, since large amounts of data are essential to conduct state-of-the-art deep learning experiments.

6 Conclusions

In this text, we explained the principles of neural machine translation. We further categorized the related work exploits external information in the translation architectures. We presented the experiments we conducted in the past and suggested ideas for future work.

In the future work proposal, we put accent on harnessing manually annotated datasets which were created at our institute in order to explore their potential in improving the translation quality.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* .

Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, Strouds-

burg, PA, USA, NAACL-Short '06, pages 1–4. http://dl.acm.org/citation.cfm?id=1614049.1614050.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2442–2452. http://www.aclweb.org/anthology/P16-1231.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating Discrete Translation Lexicons into Neural Machine Translation. *arXiv:1606.02006 [cs]* ArXiv: 1606.02006. http://arxiv.org/abs/1606.02006.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. *arXiv:1704.04675 [cs]* ArXiv: 1704.04675. http://arxiv.org/abs/1704.04675.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam M. Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems, NIPS*. http://arxiv.org/abs/1506.03099.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016a. Does Multimodality Help Human and Machine for Translation and Image Captioning? *arXiv:1605.09186 [cs]* pages 627–633. ArXiv: 1605.09186. https://doi.org/10.18653/v1/W16-2358.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016b. Multimodal Atention for Neural Machine Translation. *arXiv:1609.03976 [cs]* ArXiv: 1609.03976. http://arxiv.org/abs/1609.03976.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Incorporating global visual features into attention-based neural machine translation. *CoRR* abs/1701.06521. http://arxiv.org/abs/1701.06521.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar, pages 103–111. http://www.aclweb.org/anthology/W14-4012.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, United Kingdom, pages 85–91. http://www.aclweb.org/anthology/W11-2107.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 199–209. http://www.aclweb.org/anthology/N16-1024.

Desmond Elliott, Stella Frank, and Eva Hasler. 2015. Multi-language image description with neural sequence models. *CoRR, abs/1510.04709* .

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *CoRR* abs/1605.00459. http://arxiv.org/abs/1605.00459.

Desmond Elliott and Ákos Kádár. 2017. Imagination improves multimodal translation. *CoRR* abs/1705.04350. http://arxiv.org/abs/1705.04350.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to Parse and Translate Improves Neural Machine Translation. *arXiv:1702.03525 [cs]* ArXiv: 1702.03525. http://arxiv.org/abs/1702.03525.

Gertrud Faaß and Kerstin Eckart. 2013. Sdewac–a corpus of parsable sentences from the web. In *Language processing and knowledge in the Web*, Springer, pages 61–68.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, CA, USA, pages 866–875. http://www.aclweb.org/anthology/N16-1101.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *CoRR* abs/1410.5401. http://arxiv.org/abs/1410.5401.

Çaglar Gülçehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. *CoRR* abs/1603.00391. http://arxiv.org/abs/1603.00391.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Prague

czech-english dependency treebank 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11858/00-097C-0000-0015-8DAF-4.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural Machine Translation with Source-Side Latent Graph Parsing. *arXiv:1702.02265 [cs]* ArXiv: 1702.02265. http://arxiv.org/abs/1702.02265.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 770–778.

Jindřich Helcl and Jindřich Libovický. 2017. Neural monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics* (107):5–17. https://doi.org/10.1515/pralin-2017-0001.

Jindřich Helcl and Jindřich Libovický. 2017. Cuni system for the wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics, Copenhagen, Denmark, pages 450–457. http://www.aclweb.org/anthology/W17-4749.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv:1611.04558 [cs]* ArXiv: 1611.04558. http://arxiv.org/abs/1611.04558.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, Prague, Czech Republic, pages 177–180. http://www.aclweb.org/anthology/P07-2045.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling Source Syntax for Neural Machine Translation. *arXiv:1705.01020 [cs]* ArXiv: 1705.01020. http://arxiv.org/abs/1705.01020.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI system for WMT16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 646–654. http://www.aclweb.org/anthology/W/W16/W16-2361.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2016. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *CoRR* abs/1612.01887. http://arxiv.org/abs/1612.01887.

Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word sub-sequences and dependency sub-trees. Springer.

Jan Niehues and Eunah Cho. 2017. Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning. *arXiv:1708.00993 [cs]* ArXiv: 1708.00993. http://arxiv.org/abs/1708.00993.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, pages 392–395. http://aclweb.org/anthology/W15-3049.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. *CoRR* abs/1612.00563. http://arxiv.org/abs/1612.00563.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. *arXiv:1606.02892 [cs]* ArXiv: 1606.02892. http://arxiv.org/abs/1606.02892.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556. http://arxiv.org/abs/1409.1556.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1374–1383. http://www.aclweb.org/anthology/P13-1135.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliot. 2016. First shared task on multimodal machine translation and crosslingual image description. online. staff.fnwi.uva.nl/s.c.frank/mmt_wmt_slides.pdf.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Aleš Tamchyna, Marion Weller-Di Marco, and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation. *arXiv:1707.06012 [cs]* ArXiv: 1707.06012. http://arxiv.org/abs/1707.06012.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *CoRR* abs/1601.04811. http://arxiv.org/abs/1601.04811.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144. http://arxiv.org/abs/1609.08144.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings, Lille, France, pages 2048–2057. http://jmlr.org/proceedings/papers/v37/xuc15.pdf.

Junbei Zhang, Xiao-Dan Zhu, Qian Chen, Li-Rong Dai, Si Wei, and Hui Jiang. 2017. Exploring question understanding and adaptation in neural-network-based question answering. *CoRR* abs/1703.04617. http://arxiv.org/abs/1703.04617.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, CA, USA, pages 30–34. http://www.aclweb.org/anthology/N16-1004.