# Are BLEU and Meaning Representation in Opposition?

**Ondřej Cífka** and **Ondřej Bojar**
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{cifka,bojar}@ufal.mff.cuni.cz

## Abstract

One of possible ways of obtaining continuous-space sentence representations is by training neural machine translation (NMT) systems. The recent attention mechanism however removes the single point in the neural network from which the source sentence representation can be extracted. We propose several variations of the attentive NMT architecture bringing this meeting point back. Empirical evaluation suggests that the better the translation quality, the worse the learned sentence representations serve in a wide range of classification and similarity tasks.

## 1 Introduction

Deep learning has brought the possibility of automatically learning continuous representations of sentences. On the one hand, such representations can be geared towards particular tasks such as classifying the sentence in various aspects (e.g. sentiment, register, question type) or relating the sentence to other sentences (e.g. semantic similarity, paraphrasing, entailment). On the other hand, we can aim at "universal" sentence representations, that is representations performing reasonably well in a range of such tasks.

Regardless the evaluation criterion, the representations can be learned either in an unsupervised way (from simple, unannotated texts) or supervised, relying on manually constructed training sets of sentences equipped with annotations of the appropriate type. A different approach is to obtain sentence representations from training neural machine translation models (Hill et al., 2016).

Since Hill et al. (2016), NMT has seen substantial advances in translation quality and it is thus natural to ask how these improvements affect the learned representations.

One of the key technological changes was the introduction of "attention" (Bahdanau et al., 2014), making it even the very central component in the network (Vaswani et al., 2017). Attention allows the NMT system to dynamically choose which parts of the source are most important when deciding on the current output token. As a consequence, there is no longer a static vector representation of the sentence available in the system.

In this paper, we remove this limitation by proposing a novel encoder-decoder architecture with a structured fixed-size representation of the input that still allows the decoder to explicitly focus on different parts of the input. In other words, our NMT system has both the capacity to attend to various parts of the input and to produce static representations of input sentences.

We train this architecture on English-to-German and English-to-Czech translation and evaluate the learned representations of English on a wide range of tasks in order to assess its performance in learning "universal" meaning representations.

In Section 2, we briefly review recent efforts in obtaining sentence representations. In Section 3, we introduce a number of variants of our novel architecture. Section 4 describes some standard and our own methods for evaluating sentence representations. Section 5 then provides experimental results: translation and representation quality. The relation between the two is discussed in Section 6.

## 2 Related Work

The properties of continuous sentence representations have always been of interest to researchers working on neural machine translation. In the first works on RNN sequence-to-sequence models, Cho et al. (2014) and Sutskever et al. (2014)

*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 1362–1371
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

|  | Bahdanau et al.<br>ATTN | Sutskever et al.<br>FINAL | Cho et al.<br>FINAL-CTX | *POOL | *POOL-CTX | ATTN-CTX | Compound attention<br>ATTN-ATTN |
|---|---|---|---|---|---|---|---|
| Encoder states used | all | final | final | all | all | all | all |
| Combined using … | — | — | — | pooling | pooling | inner att. | inner att. |
| Sent. emb. available | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Dec. attends to enc. states | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Dec. attends to sent. emb. | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| Sent. emb. used in … | — | init | init+ctx | init | init+ctx | init+ctx | input for att. |

Table 1: Different RNN-based architectures and their properties. Legend: "pooling" – vectors combined by mean or max (AVGPOOL, MAXPOOL); "sent. emb." – sentence embedding, i.e. the fixed-size sentence representation computed by the encoder. "init" – initial decoder state. "ctx" – context vector, i.e. input for the decoder cell. "input for att." – input for the decoder attention.

provided visualizations of the phrase and sentence embedding spaces and observed that they reflect semantic and syntactic structure to some extent.

Hill et al. (2016) perform a systematic evaluation of sentence representation in different models, including NMT, by applying them to various sentence classification tasks and by relating semantic similarity to closeness in the representation space.

Shi et al. (2016) investigate the syntactic properties of representations learned by NMT systems by predicting sentence- and word-level syntactic labels (e.g. tense, part of speech) and by generating syntax trees from these representations.

Schwenk and Douze (2017) aim to learn language-independent sentence representations using NMT systems with multiple source and target languages. They do not consider the attention mechanism and evaluate primarily by similarity scores of the learned representations for similar sentences (within or across languages).

## 3 Model Architectures

Our proposed model architectures differ in (a) which encoder states are considered in subsequent processing, (b) how they are combined, and (c) how they are used in the decoder.

Table 1 summarizes all the examined configurations of RNN-based models. The first three (ATTN, FINAL, FINAL-CTX) correspond roughly to the standard sequence-to-sequence models, Bahdanau et al. (2014), Sutskever et al. (2014) and Cho et al. (2014), resp. The last column (ATTN-ATTN) is our main proposed architecture: compound attention, described here in Section 3.1.

In addition to RNN-based models, we experiment with the Transformer model, see Section 3.3.

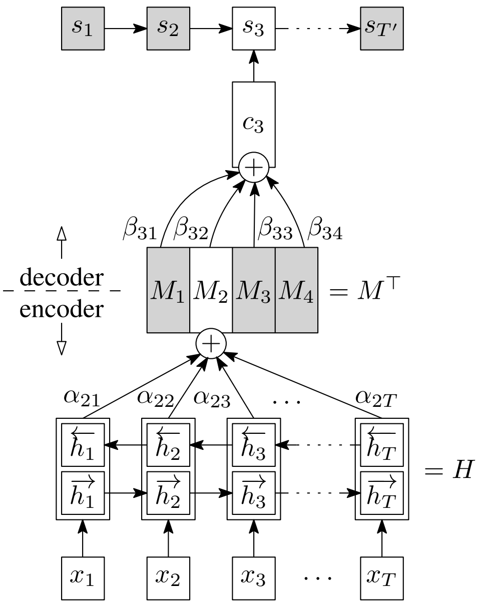

Figure 1: An illustration of compound attention with 4 attention heads. The figure shows the computations that result in the decoder state $s_3$ (in addition, each state $s_i$ depends on the previous target token $y_{i-1}$). Note that the matrix $M$ is the same for all positions in the output sentence and it can thus serve as the source sentence representation.

### 3.1 Compound Attention

Our compound attention model incorporates attention in both the encoder and the decoder, Fig. 1.

**Encoder with inner attention.** First, we process the input sequence $x_1, x_2, \ldots, x_T$ using a bi-directional recurrent network with gated recurrent units (GRU, Cho et al., 2014):

$$\begin{aligned}\overrightarrow{h_t} &= \overrightarrow{\text{GRU}}(x_t, \overrightarrow{h_{t-1}}),\\ \overleftarrow{h_t} &= \overleftarrow{\text{GRU}}(x_t, \overleftarrow{h_{t+1}}),\\ h_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}].\end{aligned}$$

We denote by $u$ the combined number of units in the two RNNs, i.e. the dimensionality of $h_t$.

Next, our goal is to combine the states $(h_1, h_2, \dots, h_T) = H$ of the encoder into a vector of fixed dimensionality that represents the entire sentence. Traditional seq2seq models concatenate the final states of both encoder RNNs ($\overrightarrow{h_T}$ and $\overleftarrow{h_1}$) to obtain the sentence representation (denoted as FINAL in Table 1). Another option is to combine all encoder states using the average or maximum over time (Collobert and Weston, 2008; Schwenk and Douze, 2017) (AVGPOOL and MAXPOOL in Table 1 and following).

We adopt an alternative approach, which is to use *inner attention*[1] (Liu et al., 2016; Li et al., 2016) to compute several weighted averages of the encoder states (Lin et al., 2017). The main motivation for incorporating these multiple "views" of the state sequence is that it removes the need for the RNN cell to accumulate the representation of the whole sentence as it processes the input, and therefore it should have more capacity for modeling local dependencies.

Specifically, we fix a number $r$, the number of *attention heads*, and compute an $r \times T$ matrix $A$ of attention weights $\alpha_{jt}$, representing the importance of position $t$ in the input for the $j^{\text{th}}$ attention head. We then use this matrix to compute $r$ weighted sums of the encoder states, which become the rows of a new matrix $M$:

$$M = AH. \qquad (1)$$

A vector representation of the source sentence (the "sentence embedding") can be obtained by flattening the matrix $M$. In our experiments, we project the encoder states $h_1, h_2, \dots, h_T$ down to a given dimensionality before applying Eq. (1), so that we can control the size of the representation.

Following Lin et al. (2017), we compute the attention matrix by feeding the encoder states to a two-layer feed-forward network:

$$A = \operatorname{softmax}(U \tanh(WH^\top)), \qquad (2)$$

where $W$ and $U$ are weight matrices of dimensions $d \times u$ and $r \times d$, respectively ($d$ is the number of hidden units); the softmax function is applied along the second dimension, i.e. across the encoder states.

**Attentive decoder.** In vanilla seq2seq models with a fixed-size sentence representation, the decoder is usually conditioned on this representation via the initial RNN state. We propose to instead leverage the structured sentence embedding by applying attention to its components. This is no different from the classical attention mechanism used in NMT (Bahdanau et al., 2014), except that it acts on this fixed-size representation instead of the sequence of encoder states.

In the $i^{\text{th}}$ decoding step, the attention mechanism computes a distribution $\{\beta_{ij}\}_{j=1}^{r}$ over the $r$ components of the structured representation. This is then used to weight these components to obtain the context vector $c_i$, which in turn is used to update the decoder state. Again, we can write this in matrix form as

$$C = BM, \qquad (3)$$

where $B = (\beta_{ij})_{i=1,j=1}^{T',r}$ is the attention matrix and $C = (c_i, c_2, \dots, c_{T'})$ are the context vectors.

Note that by combining Eqs. (1) and (3), we get

$$C = (BA)H. \qquad (4)$$

Hence, the composition of the encoder and decoder attentions (the "compound attention") defines an implicit alignment between the source and the target sequence. From this viewpoint, our model can be regarded as a restriction of the conventional attention model.

The decoder uses a conditional GRU cell (cGRU$_{\text{att}}$; Sennrich et al., 2017), which consists of two consecutively applied GRU blocks. The first block processes the previous target token $y_{i-1}$, while the second block receives the context vector $c_i$ and predicts the next target token $y_i$.

### 3.2 Constant Context

Compared to the FINAL model, the compound attention architecture described in the previous section undoubtedly benefits from the fact that the decoder is presented with information from the encoder (i.e. the context vectors $c_i$) in every decoding step. To investigate this effect, we include baseline models where we replace all context vectors $c_i$ with the entire sentence embedding (indicated by the suffix "-CTX" in Table 1). Specifically, we provide either the flattened matrix $M$ (for models with inner attention; ATTN or ATTN-CTX), the final state of the encoder (FINAL-CTX), or the result of mean- or max-pooling (*POOL-CTX) as a constant input to the decoder cell.

[1] Some papers call the same or similar approach *self-attention* or *single-time attention*.

| Name | Cl. | Train | Test | Task | Example Input and Label |
|---|---|---|---|---|---|
| MR | 2 | 11k | — | sentiment (movies) | *an idealistic love story that brings out the latent 15-year-old romantic in everyone.* (+) |
| CR | 2 | 4k | — | product review polarity | *no way to contact their customer service.* (−) |
| SUBJ | 2 | 10k | — | subjectivity | *a little weak – and it isn't that funny.* (subjective) |
| MPQA | 2 | 11k | — | opinion polarity | *was hoping* (+), *breach of the very constitution* (−) |
| SST2 | 2 | 68k | 2k | sentiment (movies) | *contains very few laughs and even less surprises* (−) |
| SST5 | 5 | 10k | 2k | sentiment (movies) | *it's worth taking the kids to.* (4) |
| TREC | 6 | 5k | 500 | question type | *What was Einstein s IQ?* (NUM) |
| MRPC | 2 | 4k | 2k | semantic equivalence | *Lawtey is not the first faith-based program in Florida's prison system. / But Lawtey is the first entire prison to take that path.* (−) |
| SNLI | 3 | 559k | 10k | natural language inference | *Two doctors perform surgery on patient. / Two surgeons are having lunch.* (contradiction) |
| SICK-E | 3 | 5k | 5k | natural language inference | *A group of people is near the ocean / A crowd of people is near the water* (entailment) |

Table 2: SentEval classification tasks. Tasks without a test set use 10-fold cross-validation.

| Name | Train | Test | Method |
|---|---|---|---|
| SICK-R | 5k | 5k | regression |
| STSB | 7k | 1k | regression |
| STS12 | — | 3k | cosine similarity |
| STS13 | — | 2k | cosine similarity |
| STS14 | — | 4k | cosine similarity |
| STS15 | — | 9k | cosine similarity |
| STS16 | — | 9k | cosine similarity |

Table 3: SentEval semantic relatedness tasks.

### 3.3 Transformer with Inner Attention

The Transformer (Vaswani et al., 2017) is a recently proposed model based entirely on feed-forward layers and attention. It consists of an encoder and a decoder, each with 6 layers, consisting of multi-head attention on the previous layer and a position-wise feed-forward network.

In order to introduce a fixed-size sentence representation into the model, we modify it by adding inner attention after the last encoder layer. The attention in the decoder then operates on the components of this representation (i.e. the rows of the matrix $M$). This variation on the Transformer model corresponds to the ATTN-ATTN column in Table 1 and is therefore denoted TRF-ATTN-ATTN.

## 4 Representation Evaluation

Continuous sentence representations can be evaluated in many ways, see e.g. Kiros et al. (2015), Conneau et al. (2017) or the RepEval workshops.[2]

We evaluate our learned representations with classification and similarity tasks from SentEval (Section 4.1) and by examining clusters of sentence paraphrase representations (Section 4.2).

### 4.1 SentEval

We perform evaluation on 10 classification and 7 similarity tasks using the SentEval[3] (Conneau et al., 2017) evaluation tool. This is a superset of the tasks from Kiros et al. (2015).

Table 2 describes the classification tasks (number of classes, data size, task type and an example) and Table 3 lists the similarity tasks. The similarity (relatedness) datasets contain pairs of sentences labeled with a real-valued similarity score. A given sentence representation model is evaluated either by learning to directly predict this score given the respective sentence embeddings ("regression"), or by computing the cosine similarity of the embeddings ("similarity") without the need of any training. In both cases, Pearson and Spearman correlation of the predictions with the gold ratings is reported.

See Dolan et al. (2004) for details on MRPC and Hill et al. (2016) for the remaining tasks.

### 4.2 Paraphrases

We also evaluate the representation of paraphrases. We use two paraphrase sources for this purpose: COCO and HyTER Networks.

COCO (Common Objects in Context; Lin et al., 2014) is an object recognition and image captioning dataset, containing 5 captions for each image. We extracted the captions from its validation set to form a set of $5 \times 5\text{k} = 25\text{k}$ sentences grouped by the source image. The average sentence length is 11 tokens and the vocabulary size is 9k types.

HyTER Networks (Dreyer and Marcu, 2014) are large finite-state networks representing a sub-

[2] https://repeval2017.github.io/

[3] https://github.com/facebookresearch/SentEval/

set of all possible English translations of 102 Arabic and 102 Chinese sentences. The networks were built by manually based on reference sentences in Arabic, Chinese and English. Each network contains up to hundreds of thousands of possible translations of the given source sentence. We randomly generated 500 translations for each source sentence, obtaining a corpus of 102k sentences grouped into 204 clusters, each containing 500 paraphrases. The average length of the 102k English sentences is 28 tokens and the vocabulary size is 11k token types.

For every model, we encode each dataset to obtain a set of sentence embeddings with cluster labels. We then compute the following metrics:

**Cluster classification accuracy** (denoted "Cl"). We remove 1 point (COCO) or half of the points (HyTER) from each cluster, and fit an LDA classifier on the rest. We then compute the accuracy of the classifier on the removed points.

**Nearest-neighbor paraphrase retrieval accuracy** (NN). For each point, we find its nearest neighbor according to cosine or $L_2$ distance, and count how often the neighbor lies in the same cluster as the original point.

**Inverse Davies-Bouldin index** (iDB). The Davies-Bouldin index (Davies and Bouldin, 1979) measures cluster separation. For every pair of clusters, we compute the ratio $R_{ij}$ of their combined scatter (average $L_2$ distance to the centroid) $S_i + S_j$ and the $L_2$ distance of their centroids $d_{ij}$, then average the maximum values for all clusters:

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \tag{5}$$

$$\mathrm{DB} = \frac{1}{N} \sum_{i=1}^{N} \max_{j \neq i} R_{ij} \tag{6}$$

The lower the DB index, the better the separation. To match with the rest of our metrics, we take its inverse: $\mathrm{iDB} = \frac{1}{\mathrm{DB}}$.

## 5 Experimental Results

We trained English-to-German and English-to-Czech NMT models using Neural Monkey[4] (Helcl and Libovický, 2017a). In the following, we distinguish these models using the code of the target language, i.e. *de* or *cs*.

The *de* models were trained on the Multi30K multilingual image caption dataset (Elliott et al., 2016), extended by Helcl and Libovický (2017b), who acquired additional parallel data using back-translation (Sennrich et al., 2016) and perplexity-based selection (Yasuda et al., 2008). This extended dataset contains 410k sentence pairs, with the average sentence length of $12 \pm 4$ tokens in English. We train each model for 20 epochs with the batch size of 32. We truecased the training data as well as all data we evaluate on. For German, we employed Neural Monkey's reversible pre-processing scheme, which expands contractions and performs morphological segmentation of determiners. We used a vocabulary of at most 30k tokens for each language (no subword units).

The *cs* models were trained on CzEng 1.7 (Bojar et al., 2016).[5] We used byte-pair encoding (BPE) with a vocabulary of 30k sub-word units, shared for both languages. For English, the average sentence length is $15 \pm 19$ BPE tokens and the original vocabulary size is 1.9M. We performed 1 training epoch with the batch size of 128 on the entire training section (57M sentence pairs).

The datasets for both *de* and *cs* models come with their respective development and test sets of sentence pairs, which we use for the evaluation of translation quality. (We use 1k randomly selected sentence pairs from CzEng 1.7 dtest as a development set. For evaluation, we use the entire etest.)

We also evaluate the InferSent model[6] (Conneau et al., 2017) as pre-trained on the natural language inference (NLI) task. InferSent has been shown to achieve state-of-the-art results on the SentEval tasks. We also include a bag-of-words baseline (GloVe-BOW) obtained by averaging GloVe[7] word vectors (Pennington et al., 2014).

### 5.1 Translation Quality

We estimate translation quality of the various models using single-reference case-sensitive BLEU (Papineni et al., 2002) as implemented in Neural Monkey (the reference implementation is `mteval-v13a.pl` from NIST or Moses).

Tables 4 and 5 provide the results on the two datasets. The *cs* dataset is much larger and the training takes much longer. We were thus able to experiment with only a subset of the possible model configurations.

[4] https://github.com/ufal/neuralmonkey

[5] http://ufal.mff.cuni.cz/czeng/czeng17

[6] https://github.com/facebookresearch/InferSent

[7] https://nlp.stanford.edu/projects/glove/

| Model | Size | Heads | BLEU dev | BLEU test |
|---|---|---|---|---|
| de-ATTN | — | — | 37.6 | 36.2 |
| de-TRF | — | — | 38.2 | 36.1 |
| de-ATTN-ATTN | 2400 | 12 | 36.2 | 34.8 |
| de-ATTN-ATTN | 1200 | 12 | 35.6 | 34.3 |
| de-ATTN-ATTN | 600 | 8 | 35.4 | 33.7 |
| de-ATTN-ATTN | 600 | 12 | 35.3 | 33.4 |
| de-ATTN-ATTN | 1200 | 6 | 35.0 | 33.2 |
| de-ATTN-ATTN | 600 | 6 | 35.1 | 33.2 |
| de-TRF-ATTN-ATTN | 600 | 3 | 32.3 | 30.1 |
| de-ATTN-ATTN | 600 | 3 | 31.4 | 29.4 |
| de-ATTN-CTX | 1200 | 12 | 30.6 | 29.2 |
| de-ATTN-CTX | 600 | 12 | 29.8 | 29.1 |
| de-ATTN-CTX | 600 | 8 | 29.8 | 28.9 |
| de-ATTN-CTX | 600 | 6 | 29.5 | 28.8 |
| de-TRF-ATTN-ATTN | 2400 | 12 | 30.6 | 28.5 |
| de-MAXPOOL-CTX | 600 | — | 27.8 | 28.1 |
| de-FINAL-CTX | 600 | — | 28.1 | 26.9 |
| de-ATTN-CTX | 600 | 3 | 27.8 | 26.9 |
| de-AVGPOOL-CTX | 600 | — | 27.1 | 26.5 |
| de-ATTN-ATTN | 600 | 1 | 27.2 | 26.0 |
| de-TRF-ATTN-ATTN | 600 | 6 | 26.5 | 25.8 |
| de-TRF-ATTN-ATTN | 1200 | 12 | 26.6 | 25.3 |
| de-FINAL | 600 | — | 23.9 | 23.8 |

Table 4: Translation quality of *de* models.

| Model | Size | Heads | BLEU dev | BLEU test | Manual > others |
|---|---|---|---|---|---|
| cs-ATTN | — | — | 22.8 | 22.2 | 89.1 |
| cs-ATTN-ATTN | 1000 | 8 | 19.1 | 18.4 | 78.8 |
| cs-ATTN-ATTN | 4000 | 4 | 18.4 | 17.9 | — |
| cs-ATTN-ATTN | 1000 | 4 | 17.5 | 17.1 | — |
| cs-ATTN-CTX | 1000 | 4 | 16.6 | 16.1 | 58.8 |
| cs-FINAL-CTX | 1000 | — | 16.1 | 15.5 | — |
| cs-ATTN-ATTN | 1000 | 1 | 15.3 | 14.8 | 49.1 |
| cs-FINAL | 1000 | — | 11.2 | 10.8 | — |
| cs-AVGPOOL | 1000 | — | 11.1 | 10.6 | — |
| cs-MAXPOOL | 1000 | — | 5.4 | 5.4 | 3.0 |

Table 5: Translation quality of *cs* models.

The columns "Size" and "Heads" specify the total size of sentence representation and the number of heads of encoder inner attention.

In both cases, the best performing is the ATTN Bahdanau et al. model, followed by Transformer (*de* only) and our ATTN-ATTN (compound attention). The non-attentive FINAL Cho et al. is the worst, except cs-MAXPOOL.

For 5 selected *cs* models, we also performed the WMT-style 5-way manual ranking on 200 sentence pairs. The annotations are interpreted as simulated pairwise comparisons. For each model, the final score is the number of times the model was judged better than the other model in the pair. Tied pairs are excluded. The results, shown in Table 5, confirm the automatic evaluation results.

We also checked the relation between BLEU and the number of heads and representation size. While there are many exceptions, the general tendency is that the larger the representation or the more heads, the higher the BLEU score. The Pearson correlation between BLEU and the number of heads is 0.87 for *cs* and 0.31 for *de*.

### 5.2 SentEval

Due to the large number of SentEval tasks, we present the results abridged in two different ways: by reporting averages (Table 6) and by showing only the best models in comparison with other methods (Table 7). The full results can be found in the supplementary material.

Table 6 provides averages of the classification and similarity results, along with the results of selected tasks (SNLI, SICK-E). As the baseline for classifications tasks, we assign the most frequent class to all test examples.[8] The *de* models are generally worse, most likely due to the higher OOV rate and overall simplicity of the training sentences. On *cs*, we see a clear pattern that more heads hurt the performance. The *de* set has more variations to consider but the results are less conclusive.

For the similarity results, it is worth noting that cs-ATTN-ATTN performs very well with 1 attention head but fails miserably with more heads. Otherwise, the relation to the number of heads is less clear.

Table 7 compares our strongest models with other approaches on all tasks. Besides InferSent and GloVe-BOW, we include SkipThought as evaluated by Conneau et al. (2017), and the NMT-based embeddings by Hill et al. (2016) trained on the English-French WMT15 dataset (this is the best result reported by Hill et al. for NMT).

We see that the supervised InferSent clearly outperforms all other models in all tasks except for MRPC and TREC. Results by Hill et al. are always lower than our best setups, except MRPC and TREC again. On classification tasks, our models are outperformed even by GloVe-BOW, except for the NLI tasks (SICK-E and SNLI) where cs-FINAL-CTX is better.

### 5.3 Paraphrase Scores

Table 6 also provides our measurements based on sentence paraphrases. For paraphrase retrieval (NN), we found cosine distance to work better

[8] For MR, CR, SUBJ, and MPQA, where there is no distinct test set, the class is established on the whole collection. For other tasks, the class is learned from the training set.

| Name | Size | H. | SNLI | SICK-E | AvgAcc | AvgSim | Hy-Cl | Hy-NN | Hy-iDB | CO-Cl | CO-NN | CO-iDB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| InferSent | 4096 | — | 83.7 | 86.4 | 81.7 | .70 | 99.99 | 100.00 | 0.579 | 31.58 | 26.21 | 0.367 |
| GloVe-BOW | 300 | — | 66.0 | 78.2 | 75.8 | .59 | 99.94 | 100.00 | 0.654 | 34.28 | 19.72 | 0.352 |
| cs-FINAL-CTX | 1000 | — | 70.2 | 82.1 | 74.4 | .60 | 99.92 | 100.00 | 0.406 | 23.20 | 16.07 | 0.346 |
| cs-ATTN-ATTN | 1000 | 1 | 69.3 | 80.8 | 73.4 | .54 | 99.88 | 99.91 | 0.347 | 21.54 | 11.50 | 0.331 |
| cs-FINAL | 1000 | — | 69.2 | 81.1 | 73.2 | .60 | 99.91 | 100.00 | 0.439 | 22.40 | 14.63 | 0.340 |
| cs-MAXPOOL | 1000 | — | 68.5 | 81.7 | 73.0 | .60 | 99.86 | 100.00 | 0.447 | 21.76 | 16.34 | 0.348 |
| cs-AVGPOOL | 1000 | — | 67.8 | 79.7 | 72.4 | .50 | 99.80 | 99.99 | 0.387 | 17.90 | 8.61 | 0.311 |
| cs-ATTN-CTX | 1000 | 4 | 66.0 | 79.5 | 72.2 | .45 | 99.75 | 99.74 | 0.287 | 14.60 | 7.54 | 0.318 |
| cs-ATTN-ATTN | 4000 | 4 | 65.2 | 78.0 | 71.2 | .39 | 99.54 | 98.98 | 0.252 | 11.52 | 5.51 | 0.303 |
| cs-ATTN-ATTN | 1000 | 4 | 64.6 | 78.0 | 70.8 | .39 | 99.26 | 98.93 | 0.253 | 10.84 | 5.20 | 0.299 |
| cs-ATTN-ATTN | 1000 | 8 | 63.2 | 76.6 | 70.0 | .36 | 99.41 | 98.09 | 0.243 | 10.24 | 4.64 | 0.287 |
| de-MAXPOOL-CTX | 600 | — | 68.0 | 78.8 | 67.1 | .50 | 98.42 | 99.90 | 0.343 | 21.54 | 15.62 | 0.341 |
| de-ATTN-CTX | 1200 | 12 | 65.0 | 77.4 | 66.7 | .52 | 98.88 | 99.91 | 0.347 | 20.06 | 16.68 | 0.348 |
| de-ATTN-CTX | 600 | 8 | 64.0 | 75.7 | 65.8 | .51 | 98.11 | 99.90 | 0.348 | 21.64 | 17.32 | 0.349 |
| de-AVGPOOL-CTX | 600 | — | 65.2 | 77.5 | 65.6 | .48 | 97.72 | 99.60 | 0.312 | 20.04 | 14.27 | 0.337 |
| de-ATTN-CTX | 600 | 12 | 61.9 | 76.0 | 65.5 | .50 | 97.79 | 99.89 | 0.360 | 20.22 | 16.10 | 0.344 |
| de-FINAL | 600 | — | 64.7 | 77.0 | 65.3 | .47 | 97.01 | 99.30 | 0.305 | 19.88 | 12.40 | 0.328 |
| de-ATTN-CTX | 600 | 3 | 63.3 | 76.0 | 65.3 | .50 | 97.81 | 99.87 | 0.328 | 19.74 | 16.43 | 0.343 |
| de-ATTN-ATTN | 600 | 1 | 63.8 | 76.9 | 64.8 | .50 | 97.70 | 99.73 | 0.352 | 19.74 | 16.26 | 0.340 |
| de-ATTN-ATTN | 600 | 3 | 61.5 | 74.7 | 64.5 | .47 | 97.42 | 99.75 | 0.314 | 17.36 | 14.35 | 0.333 |
| de-FINAL-CTX | 600 | — | 62.6 | 76.2 | 64.5 | .48 | 96.65 | 99.70 | 0.323 | 17.22 | 12.84 | 0.333 |
| de-ATTN-ATTN | 1200 | 6 | 59.6 | 72.3 | 64.3 | .41 | 98.05 | 99.80 | 0.289 | 11.90 | 10.69 | 0.327 |
| de-TRF-ATTN-ATTN | 600 | 3 | 61.4 | 72.5 | 63.9 | .49 | 95.79 | 99.64 | 0.315 | 15.76 | 14.04 | 0.340 |
| de-ATTN-ATTN | 1200 | 12 | 58.2 | 72.5 | 63.4 | .43 | 97.15 | 99.65 | 0.283 | 12.18 | 11.97 | 0.330 |
| de-ATTN-ATTN | 2400 | 12 | 59.8 | 73.9 | 63.2 | .41 | 98.69 | 99.77 | 0.287 | 10.26 | 10.94 | 0.326 |
| de-TRF-ATTN-ATTN | 2400 | 12 | 59.0 | 71.2 | 63.0 | .46 | 95.82 | 99.03 | 0.307 | 5.66 | 14.53 | 0.339 |
| de-ATTN-ATTN | 600 | 6 | 57.5 | 70.9 | 62.6 | .40 | 96.03 | 99.71 | 0.287 | 12.22 | 10.59 | 0.323 |
| de-ATTN-ATTN | 600 | 8 | 55.6 | 68.6 | 62.1 | .39 | 95.32 | 99.73 | 0.275 | 10.22 | 10.58 | 0.325 |
| de-TRF-ATTN-ATTN | 600 | 6 | 59.5 | 71.0 | 61.9 | .45 | 90.24 | 98.44 | 0.313 | 9.06 | 13.64 | 0.332 |
| de-ATTN-ATTN | 600 | 12 | 55.2 | 70.5 | 61.5 | .40 | 95.16 | 99.64 | 0.278 | 9.62 | 10.47 | 0.323 |
| de-TRF-ATTN-ATTN | 1200 | 12 | 58.2 | 68.8 | 61.1 | .46 | 90.71 | 98.22 | 0.301 | 7.06 | 13.70 | 0.333 |
| de-ATTN-CTX | 600 | 6 | 62.9 | 68.7 | 61.0 | .43 | 98.11 | 99.86 | 0.358 | 20.44 | 15.57 | 0.342 |
| LM perplexity (*cs*) | | | 190.6 | 299.4 | 1150.2 | 1224.2 | | 668.5 | | | 238.5 | |
| % OOV (*cs*) | | | 0.3 | 0.2 | 2.3 | 2.6 | | 1.2 | | | 0.1 | |
| LM perplexity (*de*) | | | 38.8 | 65.0 | 3578.2 | 2010.6 | | 3354.8 | | | 86.3 | |
| % OOV (*de*) | | | 1.5 | 1.7 | 17.8 | 16.2 | | 19.3 | | | 1.9 | |

Table 6: Abridged SentEval and paraphrase evaluation results. Full results in supplementary material. AvgAcc is the average of all 10 SentEval classification tasks (see Table S1 in supplementary material), AvgSim averages all 7 similarity tasks (see Table S2). Hy- and CO- stand for HyTER and COCO, respectively. "H." is the number of attention heads. We give the out-of-vocabulary (OOV) rate and the perplexity of a 4-gram language model (LM) trained on the English side of the respective parallel corpus and evaluated on all available data for a given task.

| Name | Size | H. | MR | CR | SUBJ | MPQA | SST2 | SST5 | TREC | MRPC | SICK-E | SNLI | AvgAcc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Most frequent baseline | | | 50.0 | 63.8 | 50.0 | 68.8 | 49.9 | 23.1 | 18.8 | 66.5 | 56.7 | 34.3 | 48.19 |
| InferSent | 4096 | — | **81.5** | **86.7** | 92.7 | **90.6** | **85.0** | **45.8** | 88.2 | 76.6 | **86.4** | (83.7) | **81.7** |
| Hill et al. en→fr† | 2400 | — | 64.7 | 70.1 | 84.9 | 81.5 | — | — | 82.8 | **96.1** | — | — | — |
| SkipThought-LN† | — | — | 79.4 | 83.1 | **93.7** | 89.3 | 82.9 | — | 88.4 | — | 79.5 | — | — |
| GloVe-BOW | 300 | — | 77.0 | 78.2 | 91.1 | 87.9 | 81.0 | 44.4 | 82.0 | 72.3 | 78.2 | 66.0 | 75.8 |
| cs-FINAL-CTX | 1000 | — | 68.7 | 77.4 | 88.5 | 85.5 | 73.0 | 38.2 | 88.6 | 71.8 | 82.1 | 70.2 | 74.4 |
| cs-ATTN-ATTN | 1000 | 1 | 68.2 | 76.0 | 86.9 | 84.9 | 72.0 | 35.7 | **89.0** | 70.7 | 80.8 | 69.3 | 73.4 |

| Name | Size | H. | SICK-R | STSB | STS12 | STS13 | STS14 | STS15 | STS16 | AvgSim |
|---|---|---|---|---|---|---|---|---|---|---|
| InferSent | 4096 | — | **.88/.83** | **.76/.75** | **.59/.60** | **.59/.59** | **.70/.67** | **.71/.72** | **.71/.73** | **.70** |
| SkipThought-LN† | — | — | .85/ — | — | — | — | .44/.45 | — | — | — |
| GloVe-BOW | 300 | — | .80/.72 | .64/.62 | .52/.53 | .50/.51 | .55/.56 | .56/.59 | .51/.58 | .59 |
| cs-FINAL-CTX | 1000 | — | .82/.76 | .74/.74 | .51/.53 | .44/.44 | .52/.50 | .62/.61 | .57/.58 | .60 |
| cs-ATTN-ATTN | 1000 | 1 | .81/.76 | .73/.73 | .46/.49 | .32/.33 | .45/.44 | .53/.52 | .47/.48 | .54 |

Table 7: Comparison of state-of-the-art SentEval results with our best models and the Glove-BOW baseline. "H." is the number of attention heads. Reprinted results are marked with †, others are our measurements.

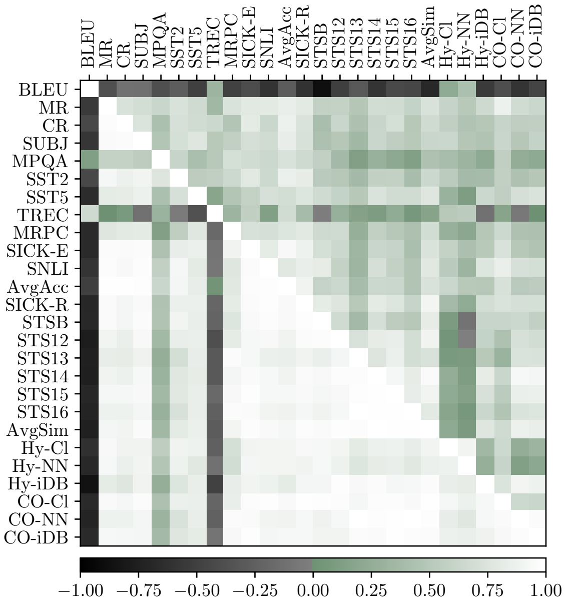

Figure 2: Pearson correlations. Upper triangle: *de* models, lower triangle: *cs* models. Positive values shown in shades of green. For similarity tasks, only the Pearson (not Spearman) coefficient is represented.

than $L_2$ distance. We therefore do not list $L_2$-based results (except in the supplementary material).

This evaluation seems less stable and discerning than the previous two, but we can again confirm the victory of InferSent followed by our non-attentive *cs* models. *cs* and *de* models are no longer clearly separated.

# 6 Discussion

To assess the relation between the various measures of sentence representations and translation quality as estimated by BLEU, we plot a heatmap of Pearson correlations in Fig. 2. As one example, Fig. 3 details the *cs* models' BLEU scores and AvgAcc.

A good sign is that on the *cs* dataset, most metrics of representation are positively correlated (the pairwise Pearson correlation is $0.78 \pm 0.32$ on average), the outlier being TREC ($-0.16 \pm 0.16$ correlation with the other metrics on average)

On the other hand, most representation metrics correlate with BLEU negatively ($-0.57 \pm 0.31$) on *cs*. The pattern is less pronounced but still clear also on the *de* dataset.

A detailed understanding of what the learned representations contain is difficult. We can only speculate that if the NMT model has some capability for following the source sentence superficially, it will use it and spend its capacity on closely matching the target sentences rather than on deriving some representation of meaning which would reflect e.g. semantic similarity. We assume that this can be a direct consequence of NMT being trained for cross entropy: putting the exact word forms in exact positions as the target sentence requires. Performing well in single-reference BLEU is not an indication that the system understands the meaning but rather that it can maximize the chance of producing the n-grams required by the reference.

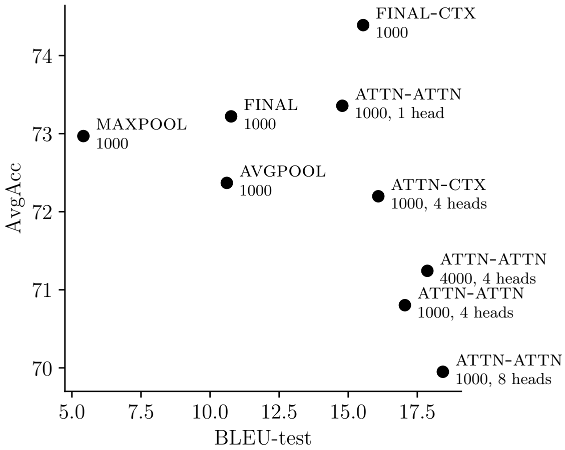

Figure 3: BLEU vs. AvgAcc for *cs* models.

The negative correlation between the number of attention heads and the representation metrics from Fig. 3 ($-0.81 \pm 0.12$ for *cs* and $-0.18 \pm 0.19$ for *de*, on average) can be partly explained by the following observation. We plotted the induced alignments (e.g. Fig. 4) and noticed that the heads tend to "divide" the sentence into segments. While one would hope that the segments correspond to some meaningful units of the sentence (e.g. subject, predicate, object), we failed to find any such interpretation for ATTN-ATTN and for *cs* models in general. Instead, the heads divide the source sentence more or less equidistantly, as documented by Fig. 5. Such a multi-headed sentence representation is then *less* fit for representing e.g. paraphrases where the subject and object swap their position due to passivization, because their representations are then accessed by different heads, and thus end up in different parts of the sentence embedding vector.

For de-ATTN-CTX models, we observed a much

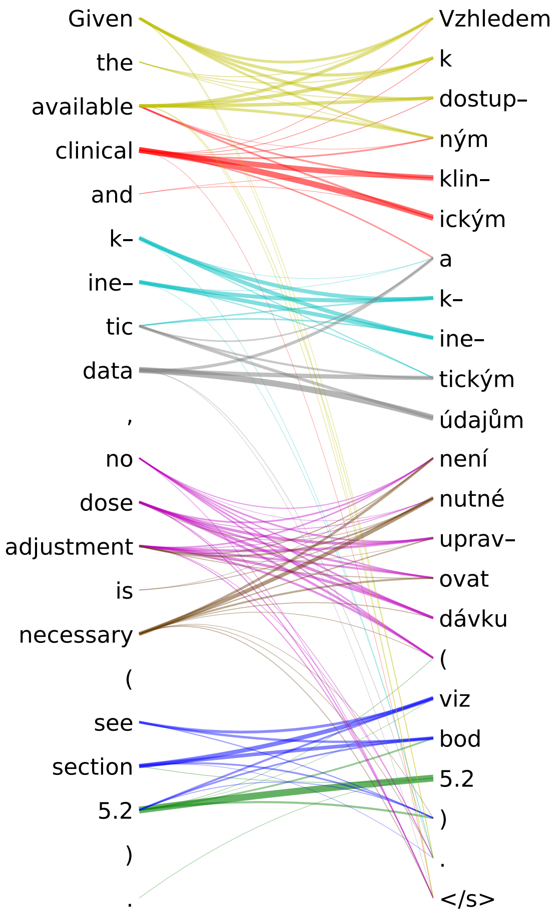

Figure 4: Alignment between a source sentence (left) and the output (right) as represented in the ATTN-ATTN model with 8 heads and size of 1000. Each color represents a different head; the stroke width indicates the alignment weight; weights ≤ 0.01 pruned out. (Best viewed in color.)

flatter distribution of attention weights for each head and, unlike in the other models, we were often able to identify a head focusing on the main verb. This difference between ATTN-ATTN and some ATTN-CTX models could be explained by the fact that in the former, the decoder is oblivious to the ordering of the heads (because of decoder attention), and hence it may not be useful for a given head to look for a specific syntactic or semantic role.

## 7 Conclusion

We presented a novel variation of attentive NMT models (Bahdanau et al., 2014; Vaswani et al., 2017) that again provides a single meeting point with a continuous representation of the source sentence. We evaluated these representations with a number of measures reflecting how well the meaning of the source sentence is captured.

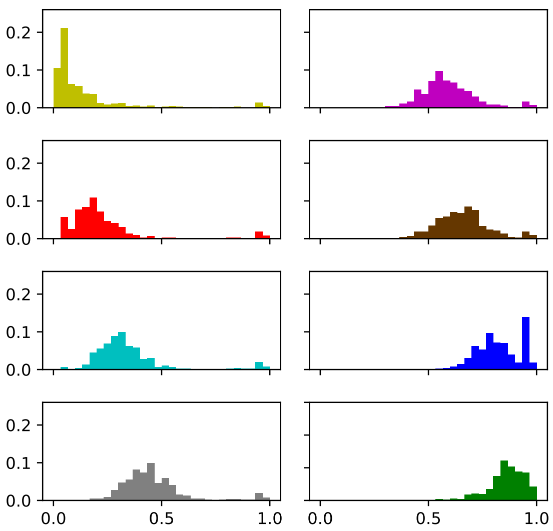

Figure 5: Attention weight by relative position in the source sentence (average over dev set excluding sentences shorter than 8 tokens). Same model as in Fig. 4. Each plot corresponds to one head.

While our proposed "compound attention" leads to translation quality not much worse than the fully attentive model, it generally does not perform well in the meaning representation. Quite on the contrary, the better the BLEU score, the worse the meaning representation.

We believe that this observation is important for representation learning where bilingual MT now seems less likely to provide useful data, but perhaps more so for MT itself, where the struggle towards a high single-reference BLEU score (or even worse, cross entropy) leads to systems that refuse to consider the meaning of the sentence.

## Acknowledgement

This work has been supported by the grants 18-24210S of the Czech Science Foundation, SVV 260 453 and "Progress" Q18+Q48 of Charles University, and using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

## References

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

Ondřej Bojar et al. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue (TSD)*, number 9924 in LNAI, pages 231–238.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227.

William B. Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING*.

Markus Dreyer and Daniel Marcu. 2014. HyTER networks of selected OpenMT08/09 sentences. Linguistic Data Consortium. LDC2014T09.

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

Jindřich Helcl and Jindřich Libovický. 2017a. Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 107(1):5–17.

Jindřich Helcl and Jindřich Libovický. 2017b. CUNI System for the WMT17 Multimodal Traslation Task.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *HLT-NAACL*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS Vol. 2*, NIPS'15, pages 3294–3302.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. 2016. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. *CoRR*, abs/1607.06275.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. volume abs/1704.04154.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich et al. 2017. Nematus: a toolkit for neural machine translation. In *EACL*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *EMNLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *IJCNLP*.