What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties

Supplementary materials

1 Appendix

Amazon Mechanical Turk survey

Subjects were recruited through the standard Amazon Mechanical Turk interface.[1] We created independent surveys for the SOMO, CoordInv and BShift tasks. We asked subjects to identify which sentences were acceptable and which were anomalous/inverted. Participants were restricted to those based in an English-speaking country.

To maximize annotation quality, we created a control set. Two authors annotated 200 random sentences from each task in a blind pretest. Those sentences on which they agreed were included in the control set.

We collected at least 10 judgments per sentence, for 1k random sentences from each task. We only retained judgments by subjects that rated at least 10 control sentences with accuracy of at least 90%. After filtering, we were left with averages of 2.5, 2.9 and 12 judgments per sentence for SOMO, CoordInv and BShift, respectively. Responses were aggregated by majority voting, before computing the final accuracies.

We did not record any personal data from subjects, and we only used the judgments in aggregated format to produce the estimated human upper bounds reported in our tables.

Further training details

Encoder training For seq2seq tasks, after hyper-parameter tuning, we chose 2-layer LSTM decoders with 512 hidden units. For NLI, we settled on a multi-layer perceptron with 100 hidden units. As is now common in NMT, we apply Byte Pair Encoding (BPE) (Sennrich, 2017) to target sentences only, with 40k codes (see Table 1 in the main text for examples of transformed target sentences). We tune dropout rate and input embedding size, picking 1024 for BiLSTMs and 512 for Gated ConvNets. We use the Adam optimizer for BiLSTMs and SGD with momentum for Gated ConvNets (after Adam gave very poor results). The encoder representation is fed to the decoder at every time step. For model selection on the validation sets, we use BLEU score[2] for NMT and AutoEncoder, perplexity for SkipThought and accuracy for Seq2Tree and NLI.

[1] `https://www.mturk.com/`

[2] MOSES multi-bleu.perl script Koehn et al. (2007)

Table 1 reports test set performance of the various architectures on the original training tasks. For NMT and Seq2Tree, we left out two random sets of 10k sentences from the training data for dev and test. The NLI dev and test sets are the ones of SNLI. Observe how results are similar for the three encoders, while, as discussed in the main text, they differ in terms of the linguistic properties their sentence embeddings are capturing. The last row of the table reports BLEU scores for our BiLSTM architecture trained with attention, showing that the architecture is on par with current NMT models, when attention is introduced. For comparison, our attention-based model obtains 37 BLEU score on the standard WMT'14 En-Fr benchmark.

Model	**En-Fr**	**En-De**	**En-Fi**	**Seq2Tree**	**NLI**
Gated ConvNet	25.9	17.0	14.2	52.3	83.5
BiLSTM-last	27.3	17.9	14.3	55.2	84.0
BiLSTM-max	27.0	18.0	14.7	53.7	85.3
BiLSTM-Att	39.1	27.2	21.9	58.4	-

Table 1: **Test results for training tasks.** Figure of merit is BLEU score for NMT and accuracy for Seq2Tree and NLI.

Probing task training The probing task results reported in the main text are obtained with a MLP that uses the Sigmoid nonlinearity, which we found to perform better than Tanh. We tune the L^2 regularization parameter, the number of hidden states (in [50, 100, 200]) and the dropout rate (in [0, 0.1, 0.2]) on the validation set of each probing task. Only for WC, which has significantly more output classes (1000) than the other tasks, we report Logistic Regression results, since they were consistently better.

Logistic regression results

Logistic regression performance approximates MLP performance (compare Table 2 here to Table 2 in the main text). This suggests that most linguistic properties can be extracted with a linear readout of the embeddings. Interestingly, if we focus on a good model-training combination, such as BiLSTM-max trained on French NMT, the tasks where the improvement from logistic regression to MLP is relatively large (>3%) are those arguably requiring the most nuanced linguistic knowledge (TreeDepth, SOMO, CoordInv).

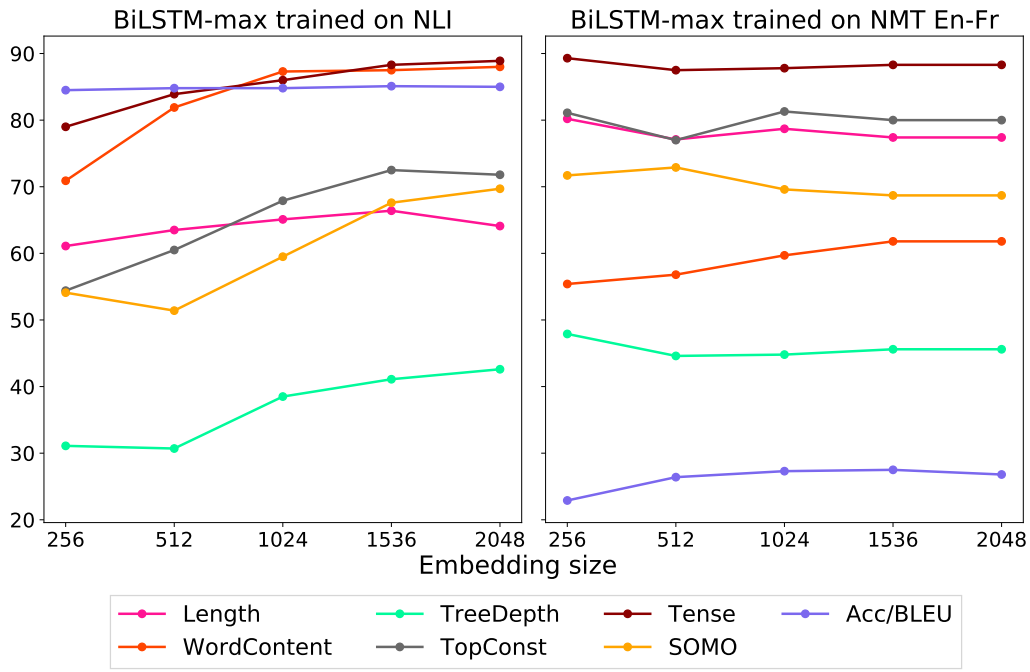

Figure 1: **Evolution of probing tasks results wrt. embedding size**. The sentence representations are generated by a BiLSTM-max encoder trained on either NLI or NMT En-Fr, with increasing sentence embedding size.

Downstream task results

We evaluate our architecture+training method combinations on the downstream tasks from the SentEval toolkit.[3] See documentation there for the tasks, that range from subjectivity analysis to question-type classification, to paraphrase detection and entailment. Also refer to the SentEval page and to Conneau et al. (2017) for the specifics of training and figures of merit for each task. In all cases, we used as input our pre-trained embeddings without fine-tuning them to the tasks. Results are reported in Table 3.

We replicate the finding of Conneau and colleagues about the effectiveness of the BiLSTM architecture with max pooling, that has also a slight edge over GatedConvNet (an architecture they did not test). As for the probing tasks, we again notice that BiLSTM-max is already effective without training, and more so than the alternative architectures.

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

Task	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
Baseline representations										
Majority vote	20.0	0.5	17.9	5.0	50.0	50.0	50.0	50.0	50.0	50.0
Hum. Eval.	100	100	84.0	84.0	98.0	85.0	88.0	86.5	81.2	85.0
Length	100	0.2	18.1	9.3	50.6	56.5	50.3	50.1	50.2	50.0
NB-uni-tfidf	22.7	97.8	24.1	41.9	49.5	77.7	68.9	64.0	38.0	50.5
NB-bi-tfidf	23.0	95.0	24.6	53.0	63.8	75.9	69.1	65.4	39.9	55.7
BoV fastText	54.8	91.6	32.3	63.1	50.8	87.8	81.9	79.3	50.3	52.7
BiLSTM-last encoder										
Untrained	32.6	43.8	24.6	74.1	52.2	83.7	82.8	71.8	49.9	64.5
AutoEncoder	98.9	23.3	28.2	72.5	60.1	80.0	81.2	76.8	50.7	62.5
NMT En-Fr	82.9	55.6	35.8	79.8	59.6	86.0	87.6	85.5	50.3	66.1
NMT En-De	82.7	53.1	35.2	80.1	58.3	86.6	88.3	84.5	50.5	66.1
NMT En-Fi	81.7	52.6	35.2	79.3	57.5	86.5	84.4	82.6	50.5	65.9
Seq2Tree	93.2	14.0	46.4	88.5	74.9	87.3	90.5	89.7	50.6	63.4
SkipThought	59.5	35.9	30.2	73.1	58.4	88.7	78.4	76.4	53.0	64.6
NLI	71.6	47.3	28.4	67.4	53.3	77.3	76.6	69.6	51.6	64.7
BiLSTM-max encoder										
Untrained	66.2	88.8	43.1	68.8	70.3	88.7	84.6	81.7	73.0	69.1
AutoEncoder	98.5	17.5	42.3	71.0	69.5	85.7	85.0	80.9	73.0	70.9
NMT En-Fr	79.3	58.3	45.7	80.5	71.2	87.8	88.1	86.3	69.9	71.8
NMT En-De	78.2	56.0	46.9	81.0	69.8	89.1	89.7	87.9	71.3	73.5
NMT En-Fi	77.5	58.3	45.8	80.5	69.7	88.2	88.9	86.1	71.9	72.8
Seq2Tree	91.8	10.3	54.6	88.7	80.0	89.5	91.8	90.7	68.6	69.8
SkipThought	59.6	35.7	42.7	70.5	73.4	90.1	83.3	79.0	70.3	70.1
NLI	65.1	87.3	38.5	67.9	63.8	86.0	78.9	78.5	59.5	64.9
GatedConvNet encoder										
Untrained	90.3	17.1	30.3	47.5	62.0	78.2	72.2	70.9	61.4	59.1
AutoEncoder	99.3	16.8	41.9	69.6	68.1	85.4	85.4	82.1	69.8	70.6
NMT En-Fr	84.3	41.3	36.9	73.8	63.7	85.6	85.7	83.8	58.8	68.1
NMT En-De	87.6	49.0	44.7	78.8	68.8	89.5	89.6	86.8	69.5	70.0
NMT En-Fi	89.1	51.5	44.1	78.6	67.2	88.7	88.5	86.3	68.3	71.0
Seq2Tree	94.5	8.7	53.1	87.4	80.9	89.6	91.5	90.8	68.3	71.6
SkipThought	73.2	48.4	40.4	76.2	71.6	89.8	84.0	79.8	68.9	68.0
NLI	70.9	29.2	38.8	59.3	66.8	80.1	77.7	72.8	69.0	69.1

Table 2: **Probing task accuracies with Logistic Regression.** Taking pre-learned sentence embeddings as input.

Interestingly, we also confirm Conneau et al.'s finding that NLI is the best source task for pre-training, despite the fact that, as we saw in the main text (Table 2 there), NMT pre-training leads to models that are capturing more linguistic properties. As they observed for downstream tasks, increasing the embedding dimension while adding capacity to the model is beneficial (see Figure 1) also for probing tasks in the case of NLI. However, it does not seem to boost the performance of the NMT En-Fr encoder.

Finally, the table also shows results from the literature recently obtained with various state-of-the-art general-purpose encoders, namely: SkipThought with layer normalization (Ba et al., 2016), InferSent (BiLSTM-max as trained on NLI by Conneau et al.) and MultiTask (Subramanian et al., 2018). A comparison of these results with ours confirms that we are testing models that do not lag much behind the state of the art.

Model	MR	CR	SUBJ	MPQA	SST-2	SST-5	TREC	MRPC	SICK-E	SICK-R	STSB
Baseline representations											
Chance	50.0	63.8	50.0	68.8	50.0	28.6	21.2	66.5	56.7	0.0	0.0
BoV fastText	78.2	80.2	91.8	88.0	82.3	45.1	83.4	74.4	78.9	82.0	70.2
BiLSTM-last encoder											
Untrained	69.7	70.2	84.8	87.0	77.2	37.6	79.6	68.5	71.6	68.2	54.8
AutoEncoder	66.0	70.7	85.7	81.1	70.0	36.2	84.0	69.9	72.2	67.6	58.3
NMT En-Fr	74.5	78.7	90.3	88.9	79.5	42.0	91.2	73.7	79.7	78.3	69.9
NMT En-De	74.8	78.4	89.8	88.7	78.8	42.3	88.0	74.1	78.8	77.5	69.3
NMT En-Fi	74.2	78.0	89.6	88.9	78.4	39.6	84.6	75.6	79.1	77.1	67.1
Seq2Tree	62.5	69.3	85.7	78.7	64.4	33.0	86.4	73.6	71.9	59.1	44.8
SkipThought	77.1	78.9	92.2	86.7	81.3	43.9	82.4	72.7	77.8	80.0	73.9
NLI	77.3	84.1	88.1	88.6	81.7	43.9	86.0	74.8	83.9	85.6	74.2
BiLSTM-max encoder											
Untrained	75.6	78.2	90.0	88.1	79.9	39.1	80.6	72.2	80.8	83.3	70.2
AutoEncoder	68.3	74.0	87.2	84.6	70.8	34.0	85.0	71.0	75.3	70.4	55.1
NMT En-Fr	76.5	81.1	91.4	89.7	77.7	42.2	89.6	75.1	79.3	78.8	68.8
NMT En-De	77.7	81.2	92.0	89.7	79.3	41.0	88.2	76.2	81.0	80.0	68.7
NMT En-Fi	77.0	81.1	91.5	90.0	80.3	43.4	87.2	75.0	81.7	80.3	69.5
Seq2Tree	65.2	74.4	88.3	80.2	66.5	31.6	85.0	72.0	74.8	65.1	36.1
SkipThought	78.0	82.8	93.0	87.3	81.5	41.9	86.8	73.2	80.0	82.0	71.5
NLI	79.2	86.7	90.0	89.8	83.5	46.4	86.0	74.5	84.5	87.5	76.6
GatedConvNet encoder											
Untrained	65.5	65.3	78.3	82.9	65.8	34.0	67.6	68.1	61.6	56.7	38.9
AutoEncoder	72.1	74.1	86.6	86.0	74.4	36.6	79.6	69.7	72.0	65.8	45.5
NMT En-Fr	74.5	78.3	88.7	88.4	76.8	38.3	86.2	72.5	77.3	73.2	60.4
NMT En-De	77.1	80.4	90.9	89.2	79.2	41.9	90.4	76.8	81.9	78.7	69.4
NMT En-Fi	76.9	82.0	91.2	90.0	78.8	41.9	90.0	76.7	81.1	79.5	70.8
Seq2Tree	65.3	73.1	85.0	79.8	63.7	31.8	81.2	72.9	74.0	58.4	30.8
SkipThought	76.0	81.7	91.5	87.2	77.9	41.5	88.8	72.3	79.5	80.0	67.8
NLI	76.7	84.7	87.4	89.1	79.2	40.9	82.0	70.8	82.0	84.7	64.4
Other sentence embedding methods											
SkipThought	79.4	83.1	93.7	89.3	82.9	-	88.4	72.4	79.5	85.8	72.1
InferSent	81.1	86.3	92.4	90.2	84.6	46.3	88.2	76.2	86.3	88.4	75.8
MultiTask	82.4	88.6	93.8	90.7	85.1	-	94.0	78.3	87.4	88.5	78.7

Table 3: **Downstream tasks results for various sentence encoder architectures pre-trained in different ways.**

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *Advances in neural information processing systems (NIPS)* .

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*. Copenhagen, Denmark, pages 670–680.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Mar-

cello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, pages 177–180.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of EACL (Short Papers)*. Valencia, Spain, pages 376–382.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*.