PARANMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations

John Wieting[1] **Kevin Gimpel**[2]
[1]Carnegie Mellon University, Pittsburgh, PA, 15213, USA
[2]Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA
jwieting@cs.cmu.edu, kgimpel@ttic.edu

Abstract

We describe PARANMT-50M, a dataset of more than 50 million English-English sentential paraphrase pairs. We generated the pairs automatically by using neural machine translation to translate the non-English side of a large parallel corpus, following Wieting et al. (2017). Our hope is that PARANMT-50M can be a valuable resource for paraphrase generation and can provide a rich source of semantic knowledge to improve downstream natural language understanding tasks. To show its utility, we use PARANMT-50M to train paraphrastic sentence embeddings that outperform all supervised systems on every SemEval semantic textual similarity competition, in addition to showing how it can be used for paraphrase generation.[1]

1 Introduction

While many approaches have been developed for generating or finding paraphrases (Barzilay and McKeown, 2001; Lin and Pantel, 2001; Dolan et al., 2004), there do not exist any freely-available datasets with millions of sentential paraphrase pairs. The closest such resource is the Paraphrase Database (PPDB; Ganitkevitch et al., 2013), which was created automatically from bilingual text by pivoting over the non-English language (Bannard and Callison-Burch, 2005). PPDB has been used to improve word embeddings (Faruqui et al., 2015; Mrkšić et al., 2016). However, PPDB is less useful for learning *sentence* embeddings (Wieting and Gimpel, 2017).

In this paper, we describe the creation of a dataset containing more than 50 million sentential paraphrase pairs. We create it automatically by scaling up the approach of Wieting et al. (2017). We use neural machine translation (NMT) to translate the Czech side of a large Czech-English parallel corpus. We pair the English translations with the English references to form paraphrase pairs. We call this dataset PARANMT-50M. It contains examples illustrating a broad range of paraphrase phenomena; we show examples in Section 3. PARANMT-50M has the potential to be useful for many tasks, from linguistically controlled paraphrase generation, style transfer, and sentence simplification to core NLP problems like machine translation.

We show the utility of PARANMT-50M by using it to train paraphrastic sentence embeddings using the learning framework of Wieting et al. (2016b). We primarily evaluate our sentence embeddings on the SemEval semantic textual similarity (STS) competitions from 2012-2016. Since so many domains are covered in these datasets, they form a demanding evaluation for a general purpose sentence embedding model.

Our sentence embeddings learned from PARANMT-50M outperform all systems in every STS competition from 2012 to 2016. These tasks have drawn substantial participation; in 2016, for example, the competition attracted 43 teams and had 119 submissions. Most STS systems use curated lexical resources, the provided supervised training data with manually-annotated similarities, and joint modeling of the sentence pair. We use none of these, simply encoding each sentence independently using our models and computing cosine similarity between their embeddings.

We experiment with several compositional architectures and find them all to work well. We find benefit from making a simple change to learning ("mega-batching") to better leverage the large training set, namely, increasing the search space

[1] Dataset, code, and embeddings are available at https://www.cs.cmu.edu/~jwieting.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 451–462
Melbourne, Australia, July 15 - 20, 2018.

of negative examples. In the supplementary, we evaluate on general-purpose sentence embedding tasks used in past work (Kiros et al., 2015; Conneau et al., 2017), finding our embeddings to perform competitively.

Finally, in Section 6, we briefly report results showing how PARANMT-50M can be used for paraphrase generation. A standard encoder-decoder model trained on PARANMT-50M can generate paraphrases that show effects of "canonicalizing" the input sentence. In other work, fully described by Iyyer et al. (2018), we used PARANMT-50M to generate paraphrases that have a specific syntactic structure (represented as the top two levels of a linearized parse tree).

We release the PARANMT-50M dataset, our trained sentence embeddings, and our code. PARANMT-50M is the largest collection of sentential paraphrases released to date. We hope it can motivate new research directions and be used to create powerful NLP models, while adding a robustness to existing ones by incorporating paraphrase knowledge. Our paraphrastic sentence embeddings are state-of-the-art by a significant margin, and we hope they can be useful for many applications both as a sentence representation function and as a general similarity metric.

2 Related Work

We discuss work in automatically building paraphrase corpora, learning general-purpose sentence embeddings, and using parallel text for learning embeddings and similarity functions.

Paraphrase discovery and generation. Many methods have been developed for generating or finding paraphrases, including using multiple translations of the same source material (Barzilay and McKeown, 2001), using distributional similarity to find similar dependency paths (Lin and Pantel, 2001), using comparable articles from multiple news sources (Dolan et al., 2004; Dolan and Brockett, 2005; Quirk et al., 2004), aligning sentences between standard and Simple English Wikipedia (Coster and Kauchak, 2011), crowdsourcing (Xu et al., 2014, 2015; Jiang et al., 2017), using diverse MT systems to translate a single source sentence (Suzuki et al., 2017), and using tweets with matching URLs (Lan et al., 2017).

The most relevant prior work uses bilingual corpora. Bannard and Callison-Burch (2005) used methods from statistical machine translation to find lexical and phrasal paraphrases in parallel text. Ganitkevitch et al. (2013) scaled up these techniques to produce the Paraphrase Database (PPDB). Our goals are similar to those of PPDB, which has likewise been generated for many languages (Ganitkevitch and Callison-Burch, 2014) since it only needs parallel text. In particular, we follow the approach of Wieting et al. (2017), who used NMT to translate the non-English side of parallel text to get English-English paraphrase pairs. We scale up the method to a larger dataset, produce state-of-the-art paraphrastic sentence embeddings, and release all of our resources.

Sentence embeddings. Our learning and evaluation setting is the same as that of our recent work that seeks to learn paraphrastic sentence embeddings that can be used for downstream tasks (Wieting et al., 2016b,a; Wieting and Gimpel, 2017; Wieting et al., 2017). We trained models on noisy paraphrase pairs and evaluated them primarily on semantic textual similarity (STS) tasks. Prior work in learning general sentence embeddings has used autoencoders (Socher et al., 2011; Hill et al., 2016), encoder-decoder architectures (Kiros et al., 2015; Gan et al., 2017), and other sources of supervision and learning frameworks (Le and Mikolov, 2014; Pham et al., 2015; Arora et al., 2017; Pagliardini et al., 2017; Conneau et al., 2017).

Parallel text for learning embeddings. Prior work has shown that parallel text, and resources built from parallel text like NMT systems and PPDB, can be used for learning embeddings for words and sentences. Several have used PPDB as a knowledge resource for training or improving embeddings (Faruqui et al., 2015; Wieting et al., 2015; Mrkšić et al., 2016). NMT architectures and training settings have been used to obtain better embeddings for words (Hill et al., 2014a,b) and words-in-context (McCann et al., 2017). Hill et al. (2016) evaluated the encoders of English-to-X NMT systems as sentence representations. Mallinson et al. (2017) adapted trained NMT models to produce sentence similarity scores in semantic evaluations.

3 The PARANMT-50M Dataset

To create our dataset, we used back-translation of bitext (Wieting et al., 2017). We used a Czech-English NMT system to translate Czech sentences

Dataset	Avg. Length	Avg. IDF	Avg. Para. Score	Vocab. Entropy	Parse Entropy	Total Size
Common Crawl	24.0±34.7	7.7±1.1	0.83±0.16	7.2	3.5	0.16M
CzEng 1.6	13.3±19.3	7.4±1.2	0.84±0.16	6.8	4.1	51.4M
Europarl	26.1±15.4	7.1±0.6	0.95±0.05	6.4	3.0	0.65M
News Commentary	25.2±13.9	7.5±1.1	0.92±0.12	7.0	3.4	0.19M

Table 1: Statistics of 100K-samples of Czech-English parallel corpora; standard deviations are shown for averages.

Reference Translation	Machine Translation
so, what's half an hour?	half an hour won't kill you.
well, don't worry. i've taken out tons and tons of guys. lots of guys.	don't worry, i've done it to dozens of men.
it's gonna be ..... classic.	yeah, sure. it's gonna be great.
greetings, all!	hello everyone!
but she doesn't have much of a case.	but as far as the case goes, she doesn't have much.
it was good in spite of the taste.	despite the flavor, it felt good.

Table 2: Example paraphrase pairs from PARANMT-50M, where each consists of an English reference translation and the machine translation of the Czech source sentence (not shown).

from the training data into English. We paired the translations with the English references to form English-English paraphrase pairs.

We used the pretrained Czech-English model from the NMT system of Sennrich et al. (2017). Its training data includes four sources: Common Crawl, CzEng 1.6 (Bojar et al., 2016), Europarl, and News Commentary. We did not choose Czech due to any particular linguistic properties. Wieting et al. (2017) found little difference among Czech, German, and French as source languages for back-translation. There were much larger differences due to data domain, so we focus on the question of domain in this section. We leave the question of investigating properties of back-translation of different languages to future work.

3.1 Choosing a Data Source

To assess characteristics that yield useful data, we randomly sampled 100K English reference translations from each data source and computed statistics. Table 1 shows the average sentence length, the average inverse document frequency (IDF) where IDFs are computed using Wikipedia sentences, and the average paraphrase score for the two sentences. The paraphrase score is calculated by averaging PARAGRAM-PHRASE embeddings (Wieting et al., 2016b) for the two sentences in each pair and then computing their cosine similarity. The table also shows the entropies of the vocabularies and constituent parses obtained using the Stanford Parser (Manning et al., 2014).[2]

Europarl exhibits the least diversity in terms of rare word usage, vocabulary entropy, and parse entropy. This is unsurprising given its formulaic and repetitive nature. CzEng has shorter sentences than the other corpora and more diverse sentence structures, as shown by its high parse entropy. In terms of vocabulary use, CzEng is not particularly more diverse than Common Crawl and News Commentary, though this could be due to the prevalence of named entities in the latter two.

In Section 5.3, we empirically compare these data sources as training data for sentence embeddings. The CzEng corpus yields the strongest performance when controlling for training data size. Since its sentences are short, we suspect this helps ensure high-quality back-translations. A large portion of it is movie subtitles which tend to use a wide vocabulary and have a diversity of sentence structures; however, other domains are included as well. It is also the largest corpus, containing over 51 million sentence pairs. In addition to providing a large number of training examples for downstream tasks, this means that the NMT system should be able to produce quality translations for this subset of its training data.

For all of these reasons, we chose the CzEng corpus to create PARANMT-50M. When doing so, we used beam search with a beam size of 12 and selected the highest scoring translation from the beam. It took over 10,000 GPU hours to back-translate the CzEng corpus. We show illustrative examples in Table 2.

3.2 Manual Evaluation

We conducted a manual analysis of our dataset in order to quantify its noise level and assess how the

[2]To mitigate sparsity in the parse entropy, we used only the top two levels of each parse tree.

Para. Score Range	# (M)	Avg. Tri. Overlap	Paraphrase 1	2	3	Fluency 1	2	3
(-0.1, 0.2]	4.0	0.00$\pm$0.0	92	6	2	1	5	94
(0.2, 0.4]	3.8	0.02$\pm$0.1	53	32	15	1	12	87
(0.4, 0.6]	6.9	0.07$\pm$0.1	22	45	33	2	9	89
(0.6, 0.8]	14.4	0.17$\pm$0.2	1	43	56	11	0	89
(0.8, 1.0]	18.0	0.35$\pm$0.2	1	13	86	3	0	97

Table 3: Manual evaluation of PARANMT-50M. 100-pair samples were drawn from five ranges of the automatic paraphrase score (first column). Paraphrase strength and fluency were judged on a 1-3 scale and counts of each rating are shown.

noise can be ameliorated with filtering. Two native English speakers annotated a sample of 100 examples from each of five ranges of the Paraphrase Score.[3] We obtained annotations for both the strength of the paraphrase relationship and the fluency of the translations.

To annotate paraphrase strength, we adopted the annotation guidelines used by Agirre et al. (2012). The original guidelines specify six classes, which we reduced to three for simplicity. We combined the top two into one category, left the next, and combined the bottom three into the lowest category. Therefore, for a sentence pair to have a rating of 3, the sentences must have the same meaning, but some unimportant details can differ. To have a rating of 2, the sentences are roughly equivalent, with some important information missing or that differs slightly. For a rating of 1, the sentences are not equivalent, even if they share minor details.

For fluency of the back-translation, we use the following: A rating of 3 means it has no grammatical errors, 2 means it has one to two errors, and 1 means it has more than two grammatical errors or is not a natural English sentence.

Table 3 summarizes the annotations. For each score range, we report the number of pairs, the mean trigram overlap score, and the number of times each paraphrase/fluency label was present in the sample of 100 pairs. There is noise but it is largely confined to the bottom two ranges which together comprise only 16% of the entire dataset. In the highest paraphrase score range, 86% of the pairs possess a strong paraphrase relationship. The annotations suggest that PARANMT-50M contains approximately 30 million strong paraphrase pairs, and that the paraphrase score is a good indicator of quality. At the low ranges, we inspected the data and found there to be many errors in the sentence alignment in the original bitext. With regards to fluency, approximately 90% of the back-translations are fluent, even at the low end of the paraphrase score range. We do see an outlier at the second-highest range of the paraphrase score, but this may be due to the small number of annotated examples.

4 Learning Sentence Embeddings

To show the usefulness of the PARANMT-50M dataset, we will use it to train sentence embeddings. We adopt the learning framework from Wieting et al. (2016b), which was developed to train sentence embeddings from pairs in PPDB. We first describe the compositional sentence embedding models we will experiment with, then discuss training and our modification ("mega-batching").

Models. We want to embed a word sequence s into a fixed-length vector. We denote the tth word in s as s_t, and we denote its word embedding by x_t. We focus on three model families, though we also experiment with combining them in various ways. The first, which we call WORD, simply averages the embeddings x_t of all words in s. This model was found by Wieting et al. (2016b) to perform strongly for semantic similarity tasks.

The second is similar to WORD, but instead of word embeddings, we average character trigram embeddings (Huang et al., 2013). We call this TRIGRAM. Wieting et al. (2016a) found this to work well for sentence embeddings compared to other n-gram orders and to word averaging.

The third family includes long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997). We average the hidden states to produce the final sentence embedding. For regularization during training, we scramble words with a small probability (Wieting and Gimpel, 2017). We also experiment with bidirectional LSTMs (BLSTM), averaging the forward and backward hidden states with no concatenation.[4]

Training. The training data is a set S of paraphrase pairs $\langle s, s' \rangle$ and we minimize a margin-

[3] Even though the similarity score lies in $[-1, 1]$, most observed scores were positive, so we chose the five ranges shown in Table 3.

[4] Unlike Conneau et al. (2017), we found this to outperform max-pooling for both semantic similarity and general sentence embedding tasks.

based loss $\ell(s, s') =$

$$\max(0, \delta - \cos(g(s), g(s')) + \cos(g(s), g(t)))$$

where g is the model (WORD, TRIGRAM, etc.), δ is the margin, and t is a "negative example" taken from a mini-batch during optimization. The intuition is that we want the two texts to be more similar to each other than to their negative examples. To select t we choose the most similar sentence in some set. For simplicity we use the mini-batch for this set, i.e.,

$$t = \underset{t':\langle t',\cdot\rangle \in S_b \setminus \{\langle s,s'\rangle\}}{\operatorname{argmax}} \cos(g(s), g(t'))$$

where $S_b \subseteq S$ is the current mini-batch.

Modification: mega-batching. By using the mini-batch to select negative examples, we may be limiting the learning procedure. That is, if all potential negative examples in the mini-batch are highly dissimilar from s, the loss will be too easy to minimize. Stronger negative examples can be obtained by using larger mini-batches, but large mini-batches are sub-optimal for optimization.

Therefore, we propose a procedure we call "mega-batching." We aggregate M mini-batches to create one mega-batch and select negative examples from the mega-batch. Once each pair in the mega-batch has a negative example, the mega-batch is split back up into M mini-batches and training proceeds. We found that this provides more challenging negative examples during learning as shown in Section 5.5. Table 6 shows results for different values of M, showing consistently higher correlations with larger M values.

5 Experiments

We now investigate how best to use our generated paraphrase data for training paraphrastic sentence embeddings.

5.1 Evaluation

We evaluate sentence embeddings using the SemEval semantic textual similarity (STS) tasks from 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016) and the STS Benchmark (Cer et al., 2017). Given two sentences, the aim of the STS tasks is to predict their similarity on a 0-5 scale, where 0 indicates the sentences are on different topics and 5 means they are completely equivalent. As our test set, we report the average Pearson's r over each year of the STS tasks from 2012-2016. We use the small (250-example) English dataset from SemEval 2017 (Cer et al., 2017) as a development set, which we call STS2017 below.

Training Corpus	WORD	TRIGRAM	LSTM
Common Crawl	80.9	80.2	79.1
CzEng 1.6	**83.6**	**81.5**	**82.5**
Europarl	78.9	78.0	80.4
News Commentary	80.2	78.2	80.5

Table 4: Pearson's $r \times 100$ on STS2017 when training on 100k pairs from each back-translated parallel corpus. CzEng works best for all models.

The supplementary material contains a description of a method to obtain a paraphrase lexicon from PARANMT-50M that is on par with that provided by PPDB 2.0. We also evaluate our sentence embeddings on a range of additional tasks that have previously been used for evaluating sentence representations (Kiros et al., 2015).

5.2 Experimental Setup

For training sentence embeddings on PARANMT-50M, we follow the experimental procedure of Wieting et al. (2016b). We use PARAGRAM-SL999 embeddings (Wieting et al., 2015) to initialize the word embedding matrix for all models that use word embeddings. We fix the mini-batch size to 100 and the margin δ to 0.4. We train all models for 5 epochs. For optimization we use Adam (Kingma and Ba, 2014) with a learning rate of 0.001. For the LSTM and BLSTM, we fixed the scrambling rate to 0.3.[5]

5.3 Dataset Comparison

We first compare parallel data sources. We evaluate the quality of a data source by using its back-translations paired with its English references as training data for paraphrastic sentence embeddings. We compare the four data sources described in Section 3. We use 100K samples from each corpus and trained 3 different models on each: WORD, TRIGRAM, and LSTM. Table 4 shows that CzEng provides the best training data for all models, so we used it to create PARANMT-50M and for all remaining experiments.

[5] As in our prior work (Wieting and Gimpel, 2017), we found that scrambling significantly improves results, even with our much larger training set. But while we previously used a scrambling rate of 0.5, we found that a smaller rate of 0.3 worked better when training on PARANMT-50M, presumably due to the larger training set.

Filtering Method	Model Avg.
Translation Score	83.2
Trigram Overlap	83.1
Paraphrase Score	**83.3**

Table 5: Pearson's $r \times 100$ on STS2017 for the best training fold across the average of WORD, TRIGRAM, and LSTM models for each filtering method.

M	WORD	TRIGRAM	LSTM
1	82.3	81.5	81.5
20	84.0	83.1	84.6
40	**84.1**	**83.4**	**85.0**

Table 6: Pearson's $r \times 100$ on STS2017 with different mega-batch sizes M.

original	sir, i'm just trying to protect.
negative examples:	
$M = 1$	i mean, colonel...
$M = 20$	i only ask that the baby be safe.
$M = 40$	just trying to survive. on instinct.
original	i'm looking at him, you know?
$M = 1$	they know that i've been looking for her.
$M = 20$	i'm keeping him.
$M = 40$	i looked at him with wonder.
original	i'il let it go a couple of rounds.
$M = 1$	sometimes the ball doesn't go down.
$M = 20$	i'll take two.
$M = 40$	i want you to sit out a couple of rounds, all right?

Table 7: Negative examples for various mega-batch sizes M with the BLSTM model.

CzEng is diverse in terms of both vocabulary and sentence structure. It has significantly shorter sentences than the other corpora, and has much more training data, so its translations are expected to be better than those in the other corpora. Wieting et al. (2017) found that sentence length was the most important factor in filtering quality training data, presumably due to how NMT quality deteriorates with longer sentences. We suspect that better translations yield better data for training sentence embeddings.

5.4 Data Filtering

Since the PARANMT-50M dataset is so large, it is computationally demanding to train sentence embeddings on it in its entirety. So, we filter the data to create a training set for sentence embeddings.

We experiment with three simple methods: (1) the length-normalized translation score from decoding, (2) trigram overlap (Wieting et al., 2017), and (3) the paraphrase score from Section 3. Trigram overlap is calculated by counting trigrams in the reference and translation, then dividing the number of shared trigrams by the total number in the reference or translation, whichever has fewer.

We filtered the back-translated CzEng data using these three strategies. We ranked all 51M+ paraphrase pairs in the dataset by the filtering measure under consideration and then split the data into tenths (so the first tenth contains the bottom 10% under the filtering criterion, the second contains those in the bottom 10-20%, etc.).

We trained WORD, TRIGRAM, and LSTM models for a single epoch on 1M examples sampled from each of the ten folds for each filtering criterion. We averaged the correlation on the STS2017 data across models for each fold. Table 5 shows the results of the filtering methods. Filtering based on the paraphrase score produces the best data for training sentence embeddings.

We randomly selected 5M examples from the top two scoring folds using paraphrase score filtering, ensuring that we only selected examples in which both sentences have a maximum length of 30 tokens.[6] These resulting 5M examples form the training data for the rest of our experiments. Note that many more than 5M pairs from the dataset are useful, as suggested by our human evaluations in Section 3.2. We have experimented with doubling the training data when training our best sentence similarity model and found the correlation increased by more than half a percentage point on average across all datasets.

5.5 Effect of Mega-Batching

Table 6 shows the impact of varying the mega-batch size M when training for 5 epochs on our 5M-example training set. For all models, larger mega-batches improve performance. There is a smaller gain when moving from 20 to 40, but all models show clear gains over $M = 1$.

Table 7 shows negative examples with different mega-batch sizes M. We use the BLSTM model and show the negative examples (nearest neighbors from the mega-batch excluding the current training example) for three sentences. Using larger mega-batches improves performance, presumably by producing more compelling negative examples for the learning procedure. This is likely more important when training on sentences than

[6] Wieting et al. (2017) found that sentence length cutoffs were effective for filtering back-translated parallel text.

	Training Data	Model	Dim.	2012	2013	2014	2015	2016
Our Work	PARANMT	WORD	300	66.2	61.8	76.2	79.3	77.5
		TRIGRAM	300	67.2	60.3	76.1	79.7	**78.3**
		LSTM	300	67.0	62.3	76.3	78.5	76.0
		LSTM	900	**68.0**	60.4	76.3	78.8	75.9
		BLSTM	900	67.4	60.2	76.1	79.5	76.5
		WORD + TRIGRAM (addition)	300	67.3	**62.8**	**77.5**	80.1	78.2
		WORD + TRIGRAM + LSTM (addition)	300	67.1	**62.8**	76.8	79.2	77.0
		WORD, TRIGRAM (concatenation)	600	67.8	62.7	77.4	**80.3**	78.1
		WORD, TRIGRAM, LSTM (concatenation)	900	67.7	**62.8**	76.9	79.8	76.8
	SimpWiki	WORD, TRIGRAM (concatenation)	600	61.8	58.4	74.4	77.0	74.0
STS Competitions		1^{st} Place System	-	64.8	62.0	74.3	79.0	77.7
		2^{nd} Place System	-	63.4	59.1	74.2	78.0	75.7
		3^{rd} Place System	-	64.1	58.3	74.3	77.8	75.7
Related Work		InferSent (AllSNLI) (Conneau et al., 2017)	4096	58.6	51.5	67.8	68.3	67.2
		InferSent (SNLI) (Conneau et al., 2017)	4096	57.1	50.4	66.2	65.2	63.5
		FastSent (Hill et al., 2016)	100	-	-	63	-	-
		DictRep (Hill et al., 2016)	500	-	-	67	-	-
		SkipThought (Kiros et al., 2015)	4800	-	-	29	-	-
		CPHRASE (Pham et al., 2015)	-	-	-	65	-	-
		CBOW (from Hill et al., 2016)	500	-	-	64	-	-
		BLEU (Papineni et al., 2002)	-	39.2	29.5	42.8	49.8	47.4
		METEOR (Denkowski and Lavie, 2014)	-	53.4	47.6	63.7	68.8	61.8

Table 8: Pearson's $r \times 100$ on the STS tasks of our models and those from related work. We compare to the top performing systems from each SemEval STS competition. Note that we are reporting the mean correlations over domains for each year rather than weighted means as used in the competitions. Our best performing overall model (WORD, TRIGRAM) is in bold.

	Dim.	Corr.
Our Work (Unsupervised)		
WORD	300	79.2
TRIGRAM	300	79.1
LSTM	300	78.4
WORD + TRIGRAM (addition)	300	79.9
WORD + TRIGRAM + LSTM (addition)	300	79.6
WORD, TRIGRAM (concatenation)	600	79.9
WORD, TRIGRAM, LSTM (concatenation)	900	79.2
Related Work (Unsupervised)		
InferSent (AllSNLI) (Conneau et al., 2017)	4096	70.6
C-PHRASE (Pham et al., 2015)		63.9
GloVe (Pennington et al., 2014)	300	40.6
word2vec (Mikolov et al., 2013)	300	56.5
sent2vec (Pagliardini et al., 2017)	700	75.5
Related Work (Supervised)		
Dep. Tree LSTM (Tai et al., 2015)		71.2
Const. Tree LSTM (Tai et al., 2015)		71.9
CNN (Shao, 2017)		78.4

Table 9: Results on STS Benchmark test set.

prior work on learning from text snippets (Wieting et al., 2015, 2016b; Pham et al., 2015).

5.6 Model Comparison

Table 8 shows results on the 2012-2016 STS tasks and Table 9 shows results on the STS Benchmark.[7] Our best models outperform all STS competition systems and all related work of which we are aware on the 2012-2016 STS datasets. Note that the large improvement over BLEU and METEOR suggests that our embeddings could be useful for evaluating machine translation output.

Models	Mean Pearson Abs. Diff.
WORD / TRIGRAM	2.75
WORD / LSTM	2.17
TRIGRAM / LSTM	2.89

Table 10: The means (over all 25 STS competition datasets) of the absolute differences in Pearson's r between each pair of models.

Overall, our individual models (WORD, TRIGRAM, LSTM) perform similarly. Using 300 dimensions appears to be sufficient; increasing dimensionality does not necessarily improve correlation. When examining particular STS tasks, we found that our individual models showed marked differences on certain tasks. Table 10 shows the mean absolute difference in Pearson's r over all 25 datasets. The TRIGRAM model shows the largest differences from the other two, both of which use word embeddings. This suggests that TRIGRAM may be able to complement the other two by providing information about words that are unknown to models that rely on word embeddings.

We experiment with two ways of combining models. The first is to define additive architectures

[7] Baseline results are from http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark, except for the unsupervised InferSent result which we computed.

Target Syntax	Paraphrase
original	with the help of captain picard, the borg will be prepared for everything.
(SBARQ(ADVP)(,)(S)(,)(SQ))	now, the borg will be prepared by picard, will it?
(S(NP)(ADVP)(VP))	the borg here will be prepared for everything.
original	you seem to be an excellent burglar when the time comes.
(S(SBAR)(,)(NP)(VP))	when the time comes, you'll be a great thief.
(S(``)(UCP)('')(NP)(VP))	"you seem to be a great burglar, when the time comes." you said.

Table 11: Syntactically controlled paraphrases generated by the SCPN trained on PARANMT-50M.

that form the embedding for a sentence by adding the embeddings computed by two (or more) individual models. All parameters are trained jointly just like when we train individual models; that is, we do not first train two simple models and add their embeddings. The second way is to define concatenative architectures that form a sentence embedding by concatenating the embeddings computed by individual models, and again to train all parameters jointly.

In Table 8 and Table 9, these combinations show consistent improvement over the individual models as well as the larger LSTM and BLSTM. Concatenating WORD and TRIGRAM results in the best performance on average across STS tasks, outperforming the best supervised systems from each year. We have released the pretrained model for these "WORD, TRIGRAM" embeddings. In addition to providing a strong baseline for future STS tasks, these embeddings offer the advantages of being extremely efficient to compute and being robust to unknown words.

We show the usefulness of PARANMT by also reporting the results of training the "WORD, TRIGRAM" model on SimpWiki, a dataset of aligned sentences from Simple English and standard English Wikipedia (Coster and Kauchak, 2011). It has been shown useful for training sentence embeddings in past work (Wieting and Gimpel, 2017). However, Table 8 shows that training on PARANMT leads to gains in correlation of 3 to 6 points compared to SimpWiki.

6 Paraphrase Generation

In addition to powering state-of-the-art paraphrastic sentence embeddings, our dataset is useful for paraphrase generation. We briefly describe two efforts in paraphrase generation here.

We have found that training an encoder-decoder model on PARANMT-50M can produce a paraphrase generation model that canonicalizes text. For this experiment, we used a bidirectional LSTM encoder and a two-layer LSTM decoder with soft attention over the encoded states (Bahdanau et al., 2015). The attention computation consists of a bilinear product with a learned parameter matrix. Table 12 shows examples of output generated by this model, showing how the model is able to standardize the text and correct grammatical errors. This model would be interesting to evaluate for automatic grammar correction as it does so without any direct supervision. Future work could also use this canonicalization to improve performance of models by standardizing inputs and removing noise from data.

original	overall, i that it's a decent buy, and am happy that i own it.
paraphrase	it's a good buy, and i'm happy to own it.
original	oh, that's a handsome women, that is.
paraphrase	that's a beautiful woman.

Table 12: Examples from our paraphrase generation model that show the ability to canonicalize text and correct grammatical errors.

PARANMT-50M has also been used for syntactically-controlled paraphrase generation; this work is described in detail by Iyyer et al. (2018). A syntactically controlled paraphrase network (SCPN) is trained to generate a paraphrase of a sentence whose constituent structure follows a provided parse template. A parse template contains the top two levels of a linearized parse tree. Table 11 shows example outputs using the SCPN. The paraphrases mostly preserve the semantics of the input sentences while changing their syntax to fit the target syntactic templates. The SCPN was used for augmenting training data and finding adversarial examples.

We believe that PARANMT-50M and future datasets like it can be used to generate rich paraphrases that improve the performance and robustness of models on a multitude of NLP tasks.

7 Discussion

One way to view PARANMT-50M is as a way to represent the learned translation model in a mono-

lingual generated dataset. This raises the question of whether we could learn an effective sentence embedding model from the original parallel text used to train the NMT system, rather than requiring the intermediate step of generating a paraphrase training set.

However, while Hill et al. (2016) and Mallinson et al. (2017) used trained NMT models to produce sentence similarity scores, their correlations are considerably lower than ours (by 10% to 35% absolute in terms of Pearson). It appears that NMT encoders form representations that do not necessarily encode the semantics of the sentence in a way conducive to STS evaluations. They must instead create representations suitable for a decoder to generate a translation. These two goals of representing sentential semantics and producing a translation, while likely correlated, evidently have some significant differences.

Our use of an intermediate dataset leads to the best results, but this may be due to our efforts in optimizing learning for this setting (Wieting et al., 2016b; Wieting and Gimpel, 2017). Future work will be needed to develop learning frameworks that can leverage parallel text directly to reach the same or improved correlations on STS tasks.

8 Conclusion

We described the creation of PARANMT-50M, a dataset of more than 50M English sentential paraphrase pairs. We showed how to use PARANMT-50M to train paraphrastic sentence embeddings that outperform supervised systems on STS tasks, as well as how it can be used for generating paraphrases for purposes of data augmentation, robustness, and even grammar correction.

The key advantage of our approach is that it only requires parallel text. There are hundreds of millions of parallel sentence pairs, and more are being generated continually. Our procedure is immediately applicable to the wide range of languages for which we have parallel text.

We release PARANMT-50M, our code, and pretrained sentence embeddings, which also exhibit strong performance as general-purpose representations for a multitude of tasks. We hope that PARANMT-50M, along with our embeddings, can impart a notion of meaning equivalence to improve NLP systems for a variety of tasks. We are actively investigating ways to apply these two new resources to downstream applications, including machine translation, question answering, and additional paraphrase generation tasks.

Acknowledgments

We thank the anonymous reviewers, the developers of Theano (Theano Development Team, 2016), the developers of PyTorch (Paszke et al., 2017), and NVIDIA Corporation for donating GPUs used in this research.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. *Proceedings of SemEval*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of*

the International Conference on Learning Representations.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

Regina Barzilay and Kathleen R McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting on Association for Computational Linguistics*, pages 50–57.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

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 the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark.

William Coster and David Kauchak. 2011. Simple English Wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2017. Learning generic sentence representations using convolutional neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2380–2390, Copenhagen, Denmark.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia.

Felix Hill, Kyunghyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. 2014a. Embedding word similarity with neural machine translation. *arXiv preprint arXiv:1412.6448*.

Felix Hill, KyungHyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. 2014b. Not all neural embeddings are born equal. *arXiv preprint arXiv:1410.0718*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8).

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Youxuan Jiang, Jonathan K. Kummerfeld, and Walter S. Lasecki. 2017. Understanding task design trade-offs in crowdsourced paraphrase collection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 103–109, Vancouver, Canada.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*, pages 3294–3302.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):342–360.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Nghia The Pham, Germán Kruszewski, Angeliki Lazaridou, and Marco Baroni. 2015. Jointly optimizing word representations for lexical and sentential tasks with the c-phrase model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP 2004*, pages 142–149, Barcelona, Spain.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Yang Shao. 2017. HCTI at SemEval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*.

Yui Suzuki, Tomoyuki Kajiwara, and Mamoru Komachi. 2017. Building a non-trivial paraphrase corpus using multiple machine translation systems. In *Proceedings of ACL 2017, Student Research Workshop*, pages 36–42, Vancouver, Canada. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*

and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016a. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016b. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations*.

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*.

John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2078–2088, Vancouver, Canada.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, Copenhagen, Denmark.

Wei Xu, Chris Callison-Burch, and William B Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.