Beyond Contrastive Learning: A Variational Generative Model for Multilingual Retrieval

John Wieting[1], **Jonathan H. Clark**[1], **William W. Cohen**[1], **Graham Neubig**[2], **and Taylor Berg-Kirkpatrick**[3]
[1]Google DeepMind
[2]Carnegie Mellon University, Pittsburgh, PA, 15213, USA
[3]University of California San Diego, San Diego, CA, 92093, USA
{jwieting,jhclark,wcohen}@google.com, gneubig@cs.cmu.edu, tberg@eng.ucsd.edu

Abstract

Contrastive learning has been successfully used for retrieval of semantically aligned sentences, but it often requires large batch sizes and carefully engineered heuristics to work well. In this paper, we instead propose a generative model for learning multilingual text embeddings which can be used to retrieve or score sentence pairs. Our model operates on parallel data in N languages and, through an approximation we introduce, efficiently encourages source separation in this multilingual setting, separating semantic information that is shared between translations from stylistic or language-specific variation. We show careful large-scale comparisons between contrastive and generation-based approaches for learning multilingual text embeddings, a comparison that has not been done to the best of our knowledge despite the popularity of these approaches. We evaluate this method on a suite of tasks including semantic similarity, bitext mining, and cross-lingual question retrieval—the last of which we introduce in this paper. Overall, our Variational Multilingual Source-Separation Transformer (VMSST) model outperforms both a strong contrastive and generative baseline on these tasks.[1]

1 Introduction

Contrastive learning is the dominant paradigm for learning text representations from parallel text (Hermann and Blunsom, 2014; Singla et al., 2018; Guo et al., 2018; Wieting et al., 2019; Feng et al., 2022). However, contrastive learning requires strong negative examples in the data and finding these negatives can be expensive in terms of compute or manual effort. In this paper, we propose a generative[2] model for learning multilingual text embeddings which encourages source separation, separating semantic information that is shared between translations from stylistic or language-specific variation. We find that by filtering this variation into separate variables, performance of the remaining representations, that encode shared semantic information, increases across all downstream tasks.

Through an approximation that greatly reduces the memory footprint of our model, we scale our model and train on 92 languages. We systematically compare our model, the Variational Multilingual Source-Separation Transformer (VMSST) to strong contrastive and generative baselines on a suite of tasks including semantic similarity, bitext mining, and question retrieval, which we introduce for the cross-lingual setting, using the same training data and architecture. We show that our model outperforms these models and is also competitive with the state-of-the-art.

We analyze VMSST with careful ablations, showing the contribution of each aspect of the model to performance. We also show that even at large batch sizes, the advantage over contrastive learning remains, especially for large models. Furthermore, we also find the learned embedding space of our model to be smoother, making it less affected by the "hubness problem" (Radovanovic et al., 2010; Radovanović et al., 2010) in representation learning, and more suitable for large-scale retrieval than the baseline methods.

To the best of our knowledge, this is the first work to systematically compare generative and contrastive models for learning multilingual embeddings on a large parallel corpus containing many languages in a carefully controlled experimental setup—despite the popularity of these approaches (Artetxe and Schwenk, 2019b; Yang et al., 2020). We carry out these experiments with both pretrained and randomly initialized models. The comparison of objective functions is an important research question due to the large amounts of multi-

[1]Code and Flax-based T5X model checkpoint available at https://github.com/google-research/google-research/tree/master/vmsst.

[2]We mean generative both in terms of text generation and as a statistical model of the joint probability distribution.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12044–12066
July 9-14, 2023

lingual text available to train models and the many uses of these models in downstream tasks. To that end, another contribution of this paper is showing these comparisons and the surprising result that contrastive objectives do not provide the overall best accuracy on downstream tasks. Moreover, our generative VMSST increasingly outperforms the contrastive model when more layers are added and when training with larger batches and more training data, suggesting that as models continue to scale in the future, this performance gap may continue to increase further motivating the use of generative approaches for learning multilingual text embeddings.

2 Related Work

There has been a number of approaches proposed for learning bilingual and multilingual text embeddings. One popular approach is contrastive learning (Hermann and Blunsom, 2014; Singla et al., 2018; Guo et al., 2018; Wieting et al., 2019; Feng et al., 2022) where translation pairs are positive examples and text from other pairs are used as negative examples. An alternative approach is to use a neural machine translation objective, where the representation from the hidden states of the encoder is used as the sentence embedding (Espana-Bonet et al., 2017; Schwenk and Douze, 2017; Artetxe and Schwenk, 2019b). Other approaches include multi-task learning approaches which often use some type of contrastive learning of parallel text to align representations among languages (Yang et al., 2020; Goswami et al., 2021), cross-lingual pretraining (Chi et al., 2022), and model distillation from a large pretrained multilingual model (Reimers and Gurevych, 2020).

An alternative approach that is more closely related to our work is generative models that separate the linguistic variation from the shared semantic information in translation pairs. Wieting et al. (2020) considered this for bitext, with each language having its own encoder and decoder parameters. This approach however does not scale, since it is not feasible to have thousands of encoders and decoders if one wants to model all of the more than 7,000 languages in the world.

3 Model

The generative process of our underlying probabilistic model and the computation graph of our training objective procedure are depicted in Figure 1 and Figure 2 respectively. In the generative story for VMSST, we first sample a semantic variable z_{sem} for the sentence. Then for each of the N languages, we sample a language-specific variable z_{l_i}. Each latent variable z is sampled from a multivariate Gaussian prior $\mathcal{N}(0, I_k)$. These variables are then fed into a decoder that samples each of the N sentences in the translation set. Each observed translation x_{l_i}, is sampled conditioned on z_{sem} and its language variable z_{l_i}. Because z_{sem} will be used to generate the sampled sentences in all languages, we expect that this variable will encode semantic, syntactic, or stylistic information that is shared in all of the translations. Conversely, the language variables z_{l_i} will handle language-specific peculiarities or specific style differences that are not central to the meaning of the translation and are therefore not contained in many of the sentences. Concretely, the likelihood function of our model can be written for a single N-way tuple of translations $x = (x_1, ..., x_N)$:

$$p(x|z_{sem}, z_{l_1}, ..., z_{l_N}) = \prod_i^N p(x_i|z_{sem}, z_{l_i})$$

In the next section, we discuss how this separation of information is encouraged during learning.

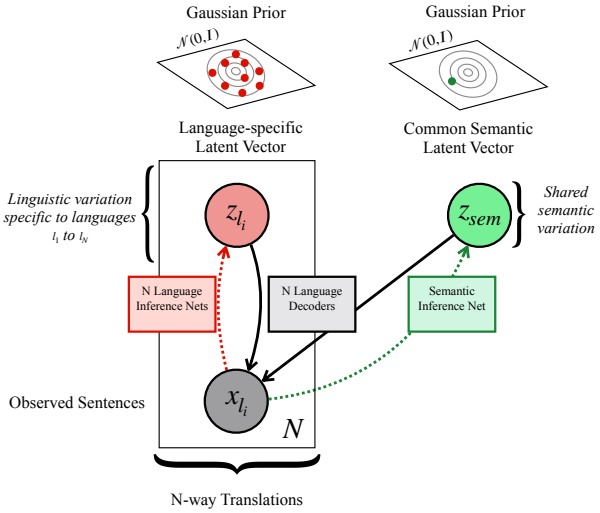

Figure 1: The generative process of our model. Latent variables, z_{l_i}, modeling the variation x_{l_i} specifically due to language l_i, as well as a latent variable modeling the common semantics, z_{sem}, are drawn from a multivariate Gaussian prior. The observed translation in each language is then conditioned on its language-specific variable and z_{sem}. In practice, we approximate this model to make learning and inference tractable.

4 Learning and Inference

We would like to train our model on a set of parallel sentences X consisting of M examples in N languages and a collection of latent variables Z. However, N-way parallel corpora are not available at the scale of bilingual text, and so we therefore approximate an N-way parallel corpus by sampling translation pairs from a large pool of pairs containing text in N languages. Therefore in our model, $X = \{\langle x_{l_i}^1, x_{l_j}^1\rangle, \ldots, \langle x_{l_i}^M, x_{l_j}^M\rangle\}$ and $Z = (\langle z_{l_i}^1, z_{l_j}^1, z_{sem}^1\rangle, \ldots, \langle z_{l_i}^M, z_{l_j}^M, z_{sem}^M\rangle)$.

We aim to maximize the likelihood of the observed X with respect to the parameters of the decoder θ, marginalizing over the latent variables Z. We follow established procedures for this optimization problem from related latent variable models like variational autoencoders (VAEs; Kingma and Welling (2013)). Specifically, we optimize a variational lower bound on the log marginal likelihood, the evidence lower bound (ELBO). ELBO introduces a variational approximation $q(z_{sem}, z_{l_i}, z_{l_j}|x_{l_i}, x_{l_j}; \phi)$ to the true posterior of the model. The q distribution is parameterized by encoders or inference networks with parameters ϕ. ELBO can be optimized by gradient ascent by using the reparameterization trick (Kingma and Welling, 2013), which allows for the expectation under q to be approximated through sampling in a way that preserves backpropagation. The decoders and encoders are discussed in further detail in Section 5.

In contrast to variational autoencoders, which have only a single latent variable for each example, we have three in our model for each example. To encourage source separation, we make several independence assumptions for q and factor it into three terms:

$$q(z_{sem}, z_{l_i}, z_{l_j}|x_{l_i}, x_{l_j}; \phi) = q(z_{sem}|x_{l_i}, x_{l_j}; \phi)q(z_{l_i}|x_{l_i}; \phi)q(z_{l_j}|x_{l_j}; \phi)$$

Lastly, we note that the ELBO contains a KL term that acts to regularize the latent variables. In our model, the KL term encourages z_{sem}, z_{l_i}, and z_{l_j} to be close to a zero-centered Gaussian prior. The KL term thus encourages source separation, as encoding information shared by the translation pair in the shared variable results in only a single penalty from the KL loss, while encoding the information separately in the language-specific variables unnecessarily doubles the overall cost. In effect, we can view these language-specific latent variables as collecting information that cannot be captured in a common semantic space, separating it out from the variables collecting shared semantic information that we use for downstream tasks.

Objective Function. The overall objective function for VMSST consists of consists of two terms, the first being ELBO as described earlier:

$$\text{ELBO} = \mathbb{E}_{q(Z_S, Z_L|X;\phi)}[\log p(X|Z_S, Z_L; \theta)] - \text{KL}(q(Z_S, Z_L|X; \phi)||p(Z_S; \theta)p(Z_L; \theta))$$

where Z_S is the collection of semantic variables, while Z_L is the collection of language variables.

The second term, which we found necessary for strong performance, is the sum of $p(x_{l_i}|\mu_{sem_{l_j}})$ and $p(x_{l_j}|\mu_{sem_{l_i}})$ which can be interpreted as samples from the mean of the posterior distribution using semantic variables generated from both input sentences. When training variational objectives, where the model ignores the latent variables and the learned posterior remains close to the prior. Examples of other approaches to address these issues include: (Yang et al., 2017; Kim et al., 2018; Xu and Durrett, 2018; He et al., 2019). We weight the ELBO by λ giving the total objective as:

$$\sum_{(x_{l_i}, x_{l_j}) \in X} p(x_{l_i}|\mu_{sem_{l_j}}) + p(x_{l_j}|\mu_{sem_{l_i}}) + \lambda \text{ELBO}$$

Therefore, our objective resembles translation with a weighted source-separation term. We show the effectiveness of this formulation compared to a pure translation objective in our experiments in Section 6.

5 Architecture

Our architecture is an encoder-decoder model, where the encoder produces a single representation that is fed into the decoder. Cross-attention between the encoder and decoder is not used, therefore the decoder has no full sequence visibility and more pressure is applied on the encoder to create a semantically meaningful representation. Specifically, we follow the approach of Wieting et al. (2020) which uses a Transformer (Vaswani et al., 2017) encoder-decoder model, where the sentence embeddings are used in two places: at each layer of the decoder in place of cross-attention and in the computation of the logits.

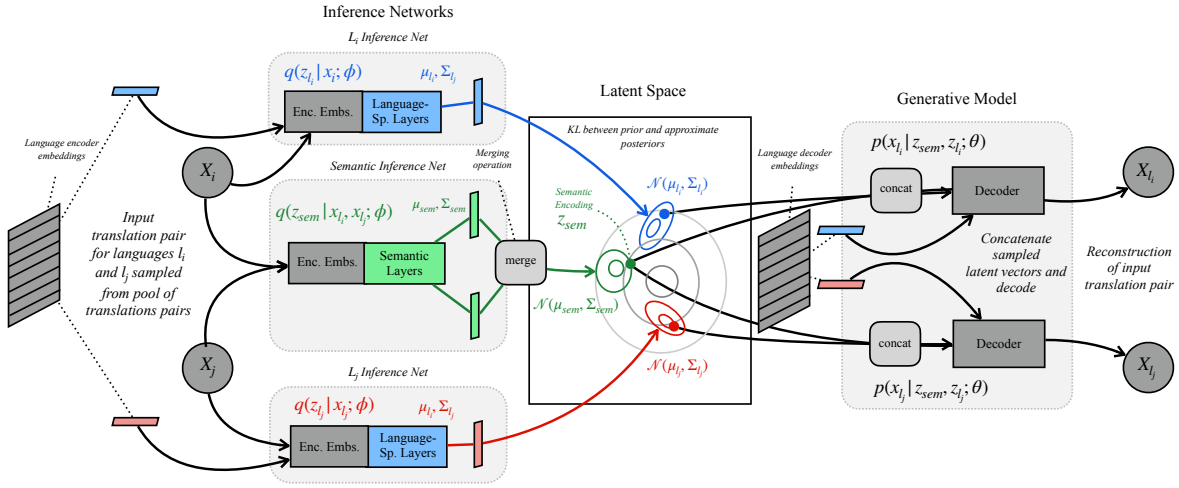

Figure 2: The computation graph for the variational lower bound used to train VMSST. The text for languages l_i and l_j, with their respective language embeddings, are fed into the encoder acting as their inference networks. The text is also fed into the semantic inference network which is a separate encoder. The output of these networks are the language variables z_{l_i} and z_{l_j} and semantic variable z_{sem}. Each language-specific variable is then concatenated to z_{sem} and used by a single shared decoder to reconstruct the input sentence pair.

Decoder Architecture. The decoder models $p(x_{l_i}|z_{sem}, z_{l_i};\theta)$ for each language i (see right side of Figure 2). The inputs to the decoder are the language-specific variable z_{l_i} and the semantic variable z_{sem}, which are concatenated and used to condition the decoder to generate the reconstruction of the observed text x_{l_i}. We use a single decoder for all languages.

Encoder Architecture. The encoders play an important role in the source separation as well as inference as detailed below.

In order to motivate the separation of the linguistic and semantic information we split the encoder into two parts, only sharing the embedding table. We use one of these encoders to be the semantic inference network, which produces the semantic variable. The other encoder represents the N language inference networks and produces the language variables for each language. These inference networks are shown on the left side of Figure 2. We mean-pool the hidden states followed by a linear projection to produce each variable from the encoders.

The semantic inference network, which models $q(z_{sem}|x_{l_i}, x_{l_j};\phi)$, is a multilingual encoder that encodes each language. For each translation pair, we alternate which of the two parallel sentences is fed into the semantic encoder within a batch for the ELBO term in the objective. Since the semantic encoder is meant to capture language agnostic semantic information, its outputs for a translation pair should be similar regardless of the language of the input sentence. We use the mean of the semantic encoder as the sentence representation for downstream tasks.

6 Experiments

6.1 Constructing the Training Data

We follow Artetxe and Schwenk (2019b) in constructing our training data. However, since the exact data is not publicly available, we expect their may be small differences due to random sampling and different dataset versions. More specifically we sample our data from Europarl,[3] United Nations (Rafalovitch and Dale, 2009),[4] OpenSubtitles2018 (Lison et al., 2018),[5] Global Voices,[6] Tanzil,[7] and Tatoeba v2021-07-22.[8]

We sample the same amount of data as was done in Artetxe and Schwenk (2019b), detailed in Appendix C. The only deviation being that we take care to not include any Tatoeba test data in our training data. Our final corpus has nearly 216 million training examples, slightly less than 220 million reported in Artetxe and Schwenk (2019b). We use both English and Spanish as pivot languages, so each pair includes at least one English or Spanish sentence, and we use approximately the same

[3] http://opus.nlpl.eu/Europarl.php
[4] https://opus.nlpl.eu/UN.php
[5] http://opus.nlpl.eu/OpenSubtitles.php
[6] https://opus.nlpl.eu/GlobalVoices.php
[7] https://opus.nlpl.eu/Tanzil.php
[8] https://opus.nlpl.eu/Tatoeba.php

amount of data for each language. We note that we only have training data for 92 languages instead of the 93 in Artetxe and Schwenk (2019b) due to not having training data for Aymara (ay).

6.2 Evaluation

We evaluate on three tasks: semantic similarity, bitext mining and question retrieval. While the first two are commonly used to evaluate multilingual sentence embeddings, we introduce question retrieval in this paper. As can be seen by our results, we found question retrieval to be somewhat uncorrelated to either of the latter two. For each task, we use a collection of different datasets, detailed below.

Semantic Textual Similarity The goal of the semantic textual similarity tasks is to predict the degree to which sentences have the same meaning as measured by human judges. The evaluation metric is Pearson's $r \times 100$ with the gold labels, which is convention for these tasks.

We make a distinction between two semantic similarity evaluations, English-only and cross-lingual. For the English-only evaluation, we follow Wieting et al. (2016) by averaging the yearly performance on 2012–2016 SemEval Semantic Textual Similarity (STS) shared tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016). More specifically, for each year of the competition, we average the Pearson's $r \times 100$ for each dataset in that year, and then finally average this result for each year of the competition. For the cross-lingual evaluation we use the cross-lingual STS tasks from SemEval 2017 (Cer et al., 2017). This evaluation contains Arabic-Arabic, Arabic-English, Spanish-Spanish, Spanish-English, and Turkish-English STS datasets. These datasets were created by translating one or both pairs of an English STS pair into Arabic (ar), Spanish (es), or Turkish (tr). We average Pearson's $r \times 100$ for these datasets.

Bitext Mining For bitext mining, we use the Tatoeba dataset introduced in Artetxe and Schwenk (2019b) and the 2018 Building and Using Parallel Corpora (BUCC) shared bitext mining task (Zweigenbaum et al., 2018).

The Tatoeba dataset consists of 100–1000 pairs of data aligned to English for 112 languages. The accuracy for Tatoeba can be computed in two ways, depending if English is the target language or source language. We compute accuracy using cosine similarity in both directions for all 112 languages (19 are unseen in the training data) and average this score for all languages.

The goal of the BUCC task is to find the gold *aligned* parallel sentences given two corpora (one being very large) in two distinct languages. Languages are aligned with English and consist of German (de), French (fr), Russian (ru), and Chinese (zh). Typically, only about 2.5% of the sentences are aligned. Following Schwenk (2018), we evaluate on the publicly available BUCC data. This involves scoring all pairs between the source target sentences and finding the optimal threshold that separates the data. Using the threshold, we can compute the precision, recall, and F_1 of the alignments. We report $F_1 \times 100$ in our results.

We compare two different approaches for finding the sentence alignments. In the first, BUCC (cosine), we compute the cosine similarity between the non-English source sentences and the English target sentences, selecting the highest scoring English sentence as the match. In the second, BUCC (margin), we follow Artetxe and Schwenk (2019a) and use a margin-based scoring approach, where the final score of a sentence pair is both a function of the score between the pair and the scores of each sentence with its nearest neighbors. To compute this margin score, we divide the cosine similarity for source sentence s_i and target sentence t_i by the sum of the scores of the four nearest neighbors of s_i with the target sentences and the sum of the scores of the four nearest neighbors of t_i with the source sentences.

Margin-based scoring is designed to alleviate the "hubness problem" (Radovanovic et al., 2010; Radovanović et al., 2010) where the neighborhood around embeddings in a high-dimensional space, like in sentence embeddings, have many neighbors in common. These neighbors can displace the correct mapping in the ordering, hurting performance.

Question Retrieval For our question retrieval evaluation, we report the accuracy (R@1) on the test sets of Natural Questions (NQ) (Kwiatkowski et al., 2019) and the Multilingual Knowledge Questions and Answers (MKQA) (Longpre et al., 2021). We use the the Probably Asked Questions dataset (PAQ) (Lewis et al., 2021) as a knowledge base from which we look up the nearest neighbor of each question in the NQ and MKQA test sets using cosine similarity. PAQ is a very large resource of 65 million automatically generated question-answer

pairs. This is a zero-shot evaluation without any NQ supervised data.[9]

Overall Score We consolidate all of these evaluations into a score, as a way to get a sense of overall performance since different models favor different evaluations. While we are averaging different metrics (accuracy, Pearson's r, and F_1), we justify this as they do have the same scale,[10] and a simple average gives a way for us to see overall performance. Our score is the average of six subtasks, two subtasks for each of semantic similarity, bitext mining, and question retrieval: English semantic similarity, cross-lingual semantic similarity, Tatoeba, BUCC (we average performance of the cosine and margin based scoring), NQ, and MKQA.

6.3 Baselines

We compare VMSST against two strong baselines, which have been used extensively in the literature.

The first baseline is CONTRASTIVE, where we use contrastive learning with the other sentences in the batch ("in-batch negative sampling") as negative examples (Sohn, 2016). CONTRASTIVE is computed as the average of computing $p(s_i|t_i)$ and $p(t_i|s_i)$ for source sentence s_i and target sentence t_i, and their respective representations $\mathbf{s_i}$ and $\mathbf{t_i}$ where the first term uses all the other targets as negatives and the second use all of the other source sentence as negatives. Specifically,

$$p(s_i|t_i) = \exp(\mathbf{s_i} \cdot \mathbf{t_i}) \; / \; \sum_{j \in B} \exp \mathbf{s_i} \cdot \mathbf{t_j}$$

$$p(t_i|s_i) = \exp(\mathbf{s_i} \cdot \mathbf{t_i}) \; / \; \sum_{j \in B} \exp \mathbf{t_i} \cdot \mathbf{s_j}$$

$$\text{loss} = -\frac{1}{2|\mathcal{B}|} \sum_{(s_i,t_i) \in \mathcal{B}} \log p(s_i|t_i) + \log p(t_i|s_i)$$

where $\mathcal{B}$ is a minibatch. This version of contrastive learning has been used in representation learning for retrieval (DPR, Karpukhin et al., 2020), visual tasks (SimCLR, Chen et al., 2020) and image/text tasks (CLIP, Radford et al., 2021). There are other variations of this loss (Qian et al., 2019), and other contrastive losses like triplet loss (Weston et al., 2010) which has been used for learning text embeddings, but we leave a comparison of contrastive objectives for learning multilingual text embeddings for future work.

The second baseline is BITRANSLATION, where we use a translation objective to learn the representation Espana-Bonet et al. (2017); Schwenk and Douze (2017); Artetxe and Schwenk (2019b).

We also explore an alternative to the VMSST, VMSST CONTRASTIVE, by incorporating a contrastive loss to use in a multitask setting. Again, we weight the contribution of the VMSST loss by λ.

6.4 Experimental Settings

We explore three different settings for each of four objective functions we consider. We use the Transformer architecture for all settings. Specifically, we explore a 6 layer encoder-decoder model, a 24 layer encoder-decoder model, and a 24 layer encoder-decoder initialized with the Multilingual T5 (mT5) Large (Xue et al., 2021). We set the dimension of the embeddings and hidden states for the encoders and decoders to 1024. The mT5 Large model inherently has embedding and hidden state dimensions of 1024. For all models, we use the mT5 vocabulary, which is derived from `sentencepiece` (Kudo and Richardson, 2018). The vocabulary consists of 250,000 tokens and was learned from multilingual variant of the C4 dataset called mC4 which includes 101 languages.

For optimization, we use Adafactor (Shazeer and Stern, 2018). We use the same learning rate schedule as Vaswani et al. (2017), i.e., the learning rate increases linearly for 4,000 steps, after which it is decayed proportionally to the inverse square root of the number of steps. We set the peak learning rate to be 0.001, and we train our models for 100,000 steps total. We use a batch size of 2048 and set the maximum sequence length of our model to 32 for all experiments.

We use a dropout rate of 0.1 for CONTRASTIVE models and no dropout for BITRANSLATION, VMSST CONTRASTIVE (with the exception of the randomly initialized 24 layer model which used 0.1), and VMSST. For VMSST, we anneal the KL term so that it increased linearly for 1,000,000 updates.

For VMSST, we set λ, the weight on the VMSST ELBO loss term, to be 0.025 for the pre-

[9] This is opposed to the formulation in the original paper where a model based on BART Large (Lewis et al., 2020a) was fine-tuned using a RAG-like objective (Lewis et al., 2020b) on the NQ training data in a model the authors call RePAQ. RePAQ, without using a reranker achieves an accuracy of 41.2 on NQ.

[10] Technically Pearson's r can be negative, but this does not happen in our evaluations.

Model	Sem. Sim.				Bitext Mining			Quest. Retrieval		Score
	Eng.	XL	XL (s.)	XL (d.)	Tatoeba	BUCC (c.)	BUCC (m.)	NQ	MKQA	
Random Init. (6 Layer)										
CONTRASTIVE	65.5	66.8	73.3	**62.4**	63.1	66.2	84.0	34.1	17.6	53.7
BITRANSLATION	69.6	63.9	71.6	58.7	53.3	62.1	81.2	**37.4**	19.2	52.5
VMSST CONTRASTIVE	65.7	66.3	73.0	61.9	**63.2**	65.8	84.3	34.1	17.7	53.7
VMSST	**70.1**	**67.4**	**75.1**	62.2	58.7	**73.7**	**85.9**	37.3	**20.1**	**55.6**
Random Init. (24 Layer)										
CONTRASTIVE	64.4	64.6	71.6	60.0	62.7	64.3	83.7	32.8	16.0	52.4
BITRANSLATION	**71.2**	68.1	74.6	63.8	57.4	70.8	86.9	38.2	21.6	55.9
VMSST CONTRASTIVE	68.2	69.7	75.5	65.9	**64.8**	58.5	84.1	36.9	18.9	55.0
VMSST	71.1	**71.7**	**77.7**	**67.7**	61.4	**78.7**	**89.0**	**38.3**	**22.3**	**58.1**
Pretrained (24 Layer)										
CONTRASTIVE	73.3	74.7	76.0	73.9	85.1	74.3	**93.7**	40.2	27.6	64.2
BITRANSLATION	74.0	78.0	79.8	76.8	78.2	85.9	91.9	**40.9**	29.6	64.9
VMSST CONTRASTIVE	73.4	75.4	76.7	74.6	**85.4**	74.6	**93.7**	40.3	27.9	64.4
VMSST	**74.6**	**79.1**	**81.5**	**77.5**	81.1	**87.8**	92.5	40.8	**29.9**	**65.9**

Table 1: Experimental results for VMSST and VMSST CONTRASTIVE and our baselines CONTRASTIVE and BITRANSLATION. We evaluate on semantic similarity, bitext mining, and question retrieval. For semantic similarity we separate the evaluations into English-only, cross-lingual, cross-lingual but with the same language (XL (s.) ar-ar and es-es) and cross-lingual using different languages (XL (d.), ar-en, es-en, and tr-en). Results are reported as the average Pearson's $r \times 100$ across datasets. For bitext mining we evaluate on Tatoeba and BUCC, with BUCC split between using cosine similarity or using a margin approach (Artetxe and Schwenk, 2019a). Results are reported as accuracy $\times 100$ for Tatoeba and $F_1 \times 100$ for BUCC. For question retrieval, we evaluate retrieval accuracy $\times 100$ using PAQ as a question knowledge base on the NQ and MKQA datasets. Finally, we compute a score to summarize quality over these evaluations.

trained models, and 0.1 when training from randomly initialized parameters. For VMSST CONTRASTIVE, we set it to .0005 for the pretrained and 6 layer settings and 0.001 for the randomly initialized 24 layer setting.

6.5 Results

The results of our experiments are shown in Table 1. Overall, VMSST has the best performance for all three experimental settings and the best performance on each task on average, with the exception of Tatoeba. In fact, for NQ question retrieval with a pretrained model, it performs nearly to that of the model trained specifically for this task on NQ data from Lewis et al. (2021) which has an accuracy of 41.2. VMSST and BITRANSLATION are especially strong when using more layers, which is not the case for CONTRASTIVE which declines in performance when moving from 6 to 24 layers. In fact at 24 layers, BITRANSLATION performs better on average than CONTRASTIVE. Perhaps for even larger models, the gap between contrastive and generative models will increase. We also see that CONTRASTIVE seems to benefit more from pretraining than VMSST and BITRANSLATION, which could possibly be due to VMSST re-purposing and adding additional randomly initialized parameters to the decoder. Perhaps different pretraining strategies using this modified decoder would resolve these differences. We also see that VMSST CONTRASTIVE has negligible improvement over CONTRASTIVE which was unexpected —that is, a traditional contrastive loss does not improve further on top of generative loss of VMSST. We leave the exploration of different strategies of combining these approaches to future work.

It is also interesting to observe the stark performance difference for different tasks. Bitext mining tasks like Tatoeba, and BUCC (m.) for the pretrained 24 layer model, favor CONTRASTIVE, while semantic similarity, BUCC (c.) and question retrieval favor VMSST, suggesting some fundamental difference in these tasks favoring CONTRASTIVE. An examination of the Tatoeba and BUCC data shows that there are paraphrases in the test set, but accounting for these does not seem to meaningfully explain this performance difference.

Lastly, we see that VMSST outperforms CONTRASTIVE on the BUCC task with cosine similarity, though the results between the two models are closer when using margin. This suggests that the "hubness problem" (Radovanovic et al., 2010; Radovanović et al., 2010) where the neighborhood around embeddings in a high-dimensional spaces

have many neighbors in common, is less of an issue when learning embeddings with VMSST. This smoother embedding space may also contribute to the stronger results VMSST has on the question retrieval tasks.

6.6 Comparison to Related Work

Prior work on learning multilingual embeddings has explored a variety of models utilizing different strategies and using difference source and types of training data. However, comparing approaches is difficult as they differ in many factors that are crucial to performance: training data, model size, architecture, vocabulary, training time, and evaluation datasets. Complicating matters further, even the metric used in evaluation for the same dataset, the distance measure used between embeddings for the same dataset, and the specific subsets of the evaluation datasets used can be different.

The main goal of this paper is to compare contrastive and generative losses systematically and uniformly, on the same data, metrics and underlying architecture. However, we also emphasize that the best systems we compare are competitive with the current state-of-the-art. Hence, in this section we compare VMSST to published results of other models on semantic similarity and the Tatoeba and BUCC bitext mining tasks. We primarily compare against five models which have the strongest multilingual results in the literature: mUSE (Yang et al., 2020), LASER (Artetxe and Schwenk, 2019b), XLM-R (NLI/STS-B) and XLM (Para.) (Reimers and Gurevych, 2020), and LaBSE (Feng et al., 2022).

For semantic similarity, we include Spearman's ρ in order to compare to work that solely uses this correlation metric. We use cosine as the similarity measure for all models in these evaluations.[11] The results are shown in Table 2.

For Tatoeba, we compare to methods that have evaluated on all 112 languages, which excludes mUSE as it was only trained on 16 language pairs. The results are shown in Table 3. Baselines results are taken from Reimers and Gurevych (2020).

For BUCC, we include results on the training sets using the margin retrieval methods from Artetxe and Schwenk (2019b). The results are shown in Table 5. Baselines results are taken from Artetxe and Schwenk (2019b); Reimers and Gurevych (2020).

Model	XL	XL (s.)	XL (d.)
mUSE	79.5	81.7	78.1
LASER	69.0	74.3	65.5
XLM-R (NLI/STS-B)	79.0	81.7	77.2
XLM-R (Para.)	**82.4**	**82.9**	**82.1**
LaBSE	72.4	74.9	70.7
VMSST	79.4	81.9	77.7

Table 2: Comparisons to related work on cross-lingual semantic similarity. Results are reported in Spearman's $\rho \times 100$. XL contains all 5 datasets, where XL (s.) contains only those where the languages are the same (ar-ar, es-es), and XL (d.) contains those datasets where the languages are different (ar-en, es-en, and tr-en). Note that models in this table are not trained on the same data; for instance LaBSE was trained on substantially more parallel data and XLM-R (Para.) was trained using a large English paraphrase corpus in addition to parallel data.

Model	Tatoeba
LASER	65.5
XLM-R (Para.)	67.1
LaBSE	**83.7**
VMSST	81.1

Table 3: Comparisons to related work on Tatoeba. Results are reported as accuracy $\times 100$, averaging the xx->en and en->xx directions. Note that models in this table are not trained on the same data; for instance LaBSE was trained on substantially more parallel data and XLM-R (Para.) was trained using a large English paraphrase corpus in addition to parallel data.

While VMSST does not have the best performance relative to models from the literature on any single task, it does have the best overall performance if one averages the results for each task.[12] While these models share much in common, namely using parallel text and some type of pretraining or pretrained model, there are differences in the exact data and models used, among other confounding variables. For instance, LaBSE used training data consisting of six billion parallel pairs across languages and was also trained on monolingual text using a masked language modelling objective. XLM-R (Para.) makes use of a 50 million example paraphrase corpus for distillation. In contrast, our setup most closely follows LASER, using an approximation of the 220M example parallel data used to train their model.

[11] Note that mUSE and LaBSE report results using the angle as the metric instead of its cosine for semantic similarity tasks, but as they do not evaluate on these specific datasets, we include the results from Reimers and Gurevych (2020) for comparison which uses cosine similarity.

[12] The average performance for VMSST is 84.3, versus 82.6 for LaBSE, and 79.3 for XLM-R (Para.)

Model	Sem. Sim.				Bitext Mining			Quest. Retrieval		Score
	Eng.	XL	XL (s.)	XL (d.)	Tatoeba	BUCC (c.)	BUCC (m.)	NQ	MKQA	
Random Init. (24 Layer)										
VMSST	71.1	**71.7**	**77.7**	**67.7**	61.4	78.7	89.0	38.3	22.3	58.1
VMSST (fact.)	67.3	69.9	76.3	65.7	**63.0**	77.9	**90.4**	37.3	21.5	57.2
VMSST (4 enc.)	**71.2**	70.2	76.6	66.0	60.8	77.7	88.5	**38.4**	22.0	57.6
VMSST (12L dec.)	71.1	70.9	77.4	66.7	61.2	78.4	88.8	38.0	22.2	57.8
VMSST (1L dec.)	71.0	71.2	77.0	67.4	**63.0**	**79.4**	89.1	38.7	**22.8**	**58.5**
VMSST (no KL)	70.7	68.7	76.2	63.7	56.9	70.8	86.6	37.8	21.5	55.7
VMSST (1 enc.)	70.6	69.4	76.7	64.6	60.0	77.0	87.8	**38.4**	21.4	57.0
VMSST (no enc. l.e.)	**71.2**	69.8	76.1	65.5	61.2	78.7	88.9	38.2	22.0	57.7
VMSST (no dec. l.e.)	70.8	70.7	76.7	66.7	60.9	77.4	88.6	38.3	21.8	57.6

Table 4: Ablations of VMSST. We investigate ablations involving factorization of the decoder projection layer (fact.), using 4 language encoders instead of 1 (4 enc.), using 12 layer (12L dec.) and 1 layer (1L dec.) decoders, using no KL term (no KL), using only a single encoder for both language and semantic variables (1 enc.), and using no encoder language embeddings (no enc. l.e.) or no decoder language embeddings (no dec. l.e.).

Model	de-en	fr-en	ru-en	zh-en	Avg.
mUSE	88.5	86.3	89.1	86.9	87.7
LASER	95.4	92.4	92.3	91.2	92.8
XLM-R (NLI/STS-B)	86.8	84.4	86.3	85.1	85.7
XLM-R (Para.)	90.8	87.1	88.6	87.8	88.6
LaBSE	**95.9**	**92.5**	**92.4**	**93.0**	**93.5**
VMSST	94.3	91.0	91.8	92.8	92.5

Table 5: Comparisons to related work on BUCC in accuracy $\times 100$ using the margin approach from Artetxe and Schwenk (2019a). Note that models in this table are not trained on the same data; for instance LaBSE was trained on substantially more parallel data and XLM-R (Para.) was trained using a large English paraphrase corpus in addition to parallel data.

7 Analysis

In this section, we analyze VMSST with additional experiments. We give a high-level overview in this section and put details and results in Appendix A.

We first investigate different ablations of the model. We analyzed aspects such as factorizing the projection layer, using weaker decoders, using 4 language-specific encoders instead of 1, removing the KL term, using a single encoder for the semantic and language embeddings, and removing the language embeddings from the encoder and from the decoder.

Secondly, we analyze the effect of the parameter sharing approximation in VMSST, where we train a full model with separate encoders and decoders for each language. This experiment uses data in 4 languages to make this experiment tractable. We found the performance to be similar enough that we can say the approximation holds, but there does remain a small gap. We hypothesize however, that this performance gap will shrink as the number of layers of the model increases.

Thirdly, we evaluate the performance of zero-shot bitext mining on languages that were unseen in the training data. We find significant improvement in this setting over the baseline BITRANSLATION. Since BITRANSLATION can be seen as an ablation of VMSST, we see that the source-separation loss especially helps with generalization to new languages.

Lastly, we investigate the impact of batch size on performance, comparing VMSST with CONTRASTIVE. It is common knowledge that contrastive models learn better representations when given harder negative examples, and bigger batch sizes increases the chances of finding these harder negatives. We experiment with batch sizes of 4096 and 8192, for both the 6 layer and 24 layer randomly initialized versions of CONTRASTIVE and VMSST. We find that both models improve when trained with larger batches, with the very best model being the 24 layer VMSST.

8 Conclusion

We present VMSST, a generative massively multilingual text embedding model trained to separate semantic information from language-specific information. VMSST also outperforms strong contrastive and generative baselines on a variety of tasks. There are several avenues for future work including alternative pretraining objectives that better fit the use case of the decoder, explore incorporating monolingual data into the generative objective, investigate synergy between VMSST and contrastive methods as they seem to specialize in different tasks, and lastly scale up to bigger models, more data, and languages to further investigate VMSST versus contrastive methods.

Limitations

Some of our experiments, specifically those in the ablations with large batch sizes, required significant computational resources. We trained these models on Google Cloud TPUv3 Pod slice with 128 chips for a few days. This experiment is important, as otherwise there would be questions on how the models compare at large batch sizes where contrastive models are known to work better. Due to training costs and in the interest of open research, we will open source our code and model checkpoints for the community to use and build upon.

Secondly, VMSST and BITRANSLATION require decoding which which means they need more memory for the decoder and are slower during training. However one advantage of these models is that they can be trained with gradient checkpointing greatly reducing their memory requirements, which cannot be used for the contrastive models as that would reduce the effective batch size for finding negative examples. Moreover, during inference, there is no difference in the memory or speed requirements in CONTRASTIVE, BITRANSLATION, or VMSST as only a single encoder is used in inference and there is no decoding.

Acknowledgements

We are grateful to Livio Baldini-Soares, Wenhu Chen, Zhuyun Dai, Tom Kwiatkowski, Jianmo Ni, Slav Petrov, Jason Riesa, and Pat Verga for useful discussions during the course of the project.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo 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)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

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)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

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. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: 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 (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

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*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo 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. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference of Machine Learning*.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182, Dublin, Ireland. Association for Computational Linguistics.

Cristina Espana-Bonet, Adám Csaba Varga, Alberto Barrón-Cedeño, and Josef van Genabith. 2017. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Koustava Goswami, Sourav Dutta, Haytham Assem, Theodorus Fransen, and John P. McCrae. 2021. Cross-lingual sentence embedding using multi-task learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9099–9113, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Brussels, Belgium. Association for Computational Linguistics.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687. PMLR.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine*

Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept):2487–2531.

Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. On the existence of obstinate results in vector space models. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193.

Alexandre Rafalovitch and Robert Dale. 2009. United Nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of Machine Translation Summit XII: Posters*, Ottawa, Canada.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234, Melbourne, Australia. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Karan Singla, Dogan Can, and Shrikanth Narayanan. 2018. A multi-task approach to learning multilingual representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 214–220, Melbourne, Australia. Association for Computational Linguistics.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, pages 5998–6008.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35.

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

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4602–4608, Florence, Italy. Association for Computational Linguistics.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A bilingual generative transformer for semantic sentence embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1581–1594, Online. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, Brussels, Belgium. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *International Conference on Machine Learning*, pages 3881–3890. JMLR. org.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2018. Overview of the third bucc shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of 11th Workshop on Building and Using Comparable Corpora*, pages 39–42.

Appendices accompanying "Beyond Contrastive Learning: A Variational Generative Model for Multilingual Retrieval"

A Analysis

A.1 Model Ablations

In this section, we investigate different ablations of VMSST. The ablations are shown in Table 4. We start from the 24 layer randomly initialized VMSST, and change it to see how certain hyperparameters and model choices affect performance.

Our first experiment, VMSST (fact.) investigates what happens if we simply factor the final projection layer of the decoder. This can save a lot of memory in the model, as that projection layer is $3 \times d \times V$ where d is the hidden state dimension size and V is the size of the vocabulary.[13] If we factor the projection layer, we can reduce the space to $d \times V + 3d \times d$. In practice, this saves about 509 million parameters for our 24 layer models. However from the first row in Table 4, we see that this small change has a significant effect on performance, weakening results on semantic similarity and question retrieval tasks and strengthening results on bitext mining tasks.

In our second ablation, VMSST (4 enc.), we spread the model capacity of the language-specific encoder to 4 encoders, instead of the single encoder in our previous experiments. We allocate the languages randomly to the different encoders. We find that this doesn't improve results, perhaps because the 24 layer model has sufficient capacity to model all of the languages in one shared encoder. We could allocate languages to encoders based on language families, and perhaps this could fare better, but we leave that for future work.

Prior work (Wieting et al., 2020) shows that, a decoder that is weaker (i.e. less layers) can lead to stronger embeddings. This effect is presumably because there is more pressure on the sentence embedding to fully and clearly capture the semantics since it cannot rely on a strong decoder to fill in gaps. We found that that using a weaker single layer decoder (1L dec.), does indeed seem to improve performance. We also tried a 12 layer ablation (12L dec.), but that seemed to not have a significant improvement in the results.

[13] We multiply by 3 because we have three embeddings, the hidden state, the language-specific vector, and the semantic vector.

The last four ablations investigate different modelling choices. In the first we eliminate the KL term (no KL), which has the most significant effect on performance, especially on cross-lingual tasks. In the second ablation, we use a single encoder instead of the twin encoders (1 enc.), one for semantic embeddings and one for language embeddings, we find that this has a modest overall effect on performance. Lastly, we eliminate the language embeddings. First we remove the language embedding inputs to the decoder (no enc. l.e.), then we experiment by removing the input language embeddings to the language-specific encoder (no dec. l.e.). We find these language embeddings have a smaller than expected impact on performance, perhaps because the large capacity of the decoder can ascertain the language being input or decoded.

A.2 Testing the Parameter Sharing in VMSST

Parameter sharing was needed in order efficiently perform source separation on N languages. Specifically we collapsed the language encoders into a single encoder and we collapsed the decoders into a single decoder. The VMSST approximates having N language encoders by using an input embedding to indicate the language being considered. The same strategy is applied with the decoders as well, with the first input token to the decoder indicating the language to be generated.

In this section, we investigate what effect this parameter sharing has on VMSST by using N encoders and decoders (full enc, full dec.). We experiment with 6 layer Transformer encoders and 4 languages Spanish, English, Arabic, and Turkish in order to keep the experiments tractable as in this setting we have 5 encoders and 4 decoders. The results are shown in Table 6.

The results indicate that the approximation appears to hold, as VMSST is much closer to the full model than BITRANSLATION, which is an ablation of VMSST without the source separation. However, there is still a gap between the full encoder/decoder of VMSST and VMSST. We hypothesize however, that as the number of layers of the model increases, this performance gap also shrinks. The extra capacity of these layers will allow for the model to separate language-specific variations without having separate parameters for each language. Evidence for this hypothesis is in Table 4 where having the language variation shared

Model	Sem. Sim.						Bitext Mining			Quest. Retrieval				Score
	Eng.	XL					Tatoeba			NQ	MKQA			
-	-	ar-en	ar-ar	es-en	es-es	tr-en	ar	es	tr	-	ar	es	tr	
Random Init. (6 Layer) - ar, en, es, tr														
CONTRASTIVE	68.6	81.7	**68.5**	**68.3**	**64.8**	**69.7**	97.9	**88.2**	**98.1**	36.4	24.6	13.0	22.6	58.1
BITRANSLATION	69.0	82.4	63.1	57.8	58.7	67.3	97.6	84.3	96.4	37.8	25.4	12.2	21.1	57.0
VMSST	**70.6**	**83.1**	65.9	63.1	62.1	68.8	97.9	84.9	97.2	38.1	**27.2**	14.4	24.1	58.5
VMSST (full enc., full dec.)	70.3	82.3	65.6	60.0	62.8	67.9	**98.4**	87.9	97.8	**39.0**	**27.2**	**14.7**	**24.2**	**58.7**

Table 6: Comparison of VMSST with a variation that has no parameter sharing, VMSST (full enc., full dec.). We experiment on 4 languages, so we have 5 encoders and 4 decoders.

amongst 4 encoders instead of 1 actually appears to weaken performance overall.

A.3 Zero-Shot Bitext Mining

The Tatoeba dataset contains parallel sentence pairs of English with 112 languages. Our model is trained using 93 of these languages, and therefore there are 19 languages we can use for a zero-shot evaluation of bitext mining. Table 8 summarizes the results of this zero-shot evaluation for the two generation objectives, BITRANSLATION and VMSST considered in this paper. The results are shown in Table 8. We also compute Δ which is the difference between the performance gap of VMSST and BITRANSLATION on the seen and unseen languages. From the results, we see that VMSST does even better than BITRANSLATION on unseen languages than unseen languages. Since BITRANSLATION can be seen as an ablation of VMSST, i.e. VMSST without the source-separation loss, we see that the source-separation loss especially helps with generalization to new languages.

A.4 Effects of Batch Size

Lastly, we investigate how VMSST compares to CONTRASTIVE as batch size increases. It is common knowledge that contrastive models learn better representations when given harder negative examples. Since we are using in-batch negatives in our contrastive baseline, the increased batch size increases the chances of encountering harder negative examples and will generally increase performance up to the point where the negatives become false. Furthermore. bigger batch sizes are known to also improve results in models using the Transformer architecture, presumably due to less noisy gradients, which would improve the results of both CONTRASTIVE and VMSST. It is important to note that using bigger batch sizes, means seeing more examples (100,000 steps at a batch size of 2048 is about 1 pass through the data). However, parallel data is so numerous that training to convergence on the available data is not very practical. Therefore, these experiments do not separate out the gains from using a bigger batch size versus seeing more training data, but we argue that is not an important distinction to make due to the sheer amount (billions of pairs) of parallel data available.

We experiment with batch sizes of 4096 and 8192, double and quadruple the 2048 used in all experiments up to this point, for both the 6 layer and 24 layer randomly initialized versions of CONTRASTIVE and VMSST. All models are trained again for 100,000 steps. The results are shown in Table 7.

From the results, we see that for the 6 layer model, increasing the batch size equalizes VMSST and CONTRASTIVE overall, however each performs better at different tasks. CONTRASTIVE has better performance on Tatoeba, XL semantic similarity, and BUCC with margin (Artetxe and Schwenk, 2019a), where VMSST has better performance on English semantic similarity, BUCC with cosine similarity, and the retrieval tasks. For the 24 layer variations, VMSST is better at every task, with the exception of Tatoeba, and has the highest overall score of any model in the table. The 24 layer CONTRASTIVE variation does not perform as well as the 6 layer version at any batch size, in contrast to VMSST where the 24 layer model always outperforms the 6 layer variation.

B Full Experimental Results

We include full results for our models using the pre-trained mT5 large checkpoint. We evaluate on English semantic similarity, Cross-lingual semantic similarity, question retrieval, and bitext mining.

Model	B. Size	Sem. Sim.				Bitext Mining			Quest. Retrieval		Score
		Eng.	XL	XL (s.)	XL (d.)	Tatoeba	BUCC (c.)	BUCC (m.)	NQ	MKQA	
Random Init. (6 Layer)											
CONTRASTIVE	2048	65.5	66.8	73.3	62.4	63.1	64.7	82.9	34.0	17.6	53.5
	4096	67.5	69.3	75.4	65.3	66.0	71.5	87.0	35.3	19.2	56.1
	8192	69.4	**71.6**	**76.8**	**68.1**	**68.6**	76.2	**89.4**	36.4	20.9	**58.3**
VMSST	2048	70.1	67.4	75.1	62.2	58.7	72.6	84.7	37.2	20.2	55.4
	4096	70.2	67.4	75.3	62.1	58.5	73.1	86.0	38.2	20.3	55.7
	8192	**71.4**	70.9	76.6	67.1	61.8	**77.9**	88.0	**39.0**	**22.4**	58.1
Random Init. (24 Layer)											
CONTRASTIVE	2048	64.4	64.6	71.6	60.0	62.7	62.8	82.5	32.8	16.0	52.2
	4096	66.6	68.6	75.1	64.3	65.7	70.9	86.8	34.7	18.1	55.4
	8192	68.0	70.2	76.2	66.2	**67.7**	74.2	88.3	35.2	19.4	57.0
VMSST	2048	71.1	71.7	77.7	67.7	61.4	78.4	87.8	38.3	22.3	58.0
	4096	72.0	72.1	77.7	68.3	62.9	81.0	89.7	38.7	23.5	59.1
	8192	**72.7**	**74.1**	**79.0**	**70.8**	64.1	**82.0**	**90.2**	**39.0**	**24.3**	**60.1**

Table 7: Comparison of CONTRASTIVE and VMSST using different batch sizes during training.

Model	Tat. (seen)	Tat. (unseen)	Δ
Random Init. (6 Layer)			
BITRANSLATION	59.3	24.0	-
VMSST	**64.4**	**30.6**	1.5
Random Init. (24 Layer)			
BITRANSLATION	82.6	56.5	-
VMSST	**84.9**	**62.2**	3.4
Pretrained (24 Layer)			
BITRANSLATION	63.7	26.5	-
VMSST	**67.3**	**32.6**	2.5

Table 8: Results on languages seen during training (seen) and languages that were not seen during training (unseen) on the Tatoeba dataset.

B.1 Semantic Similarity

For English semantic similarity, we use the SemEval semantic textual similarity (STS) tasks from 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016) as was done initially for sentence embeddings in (Wieting et al., 2016). As our test set, we report the average Pearson's r over each year of the STS tasks from 2012-2016 as is convention in the top part of Table 9. However, some recent work, like Reimers and Gurevych (2019) computed Spearman's ρ over concatenated datasets for each year of the STS competition. To be consistent with these works, we also include evaluations using this approach in the bottom part of Table 9. One other difference between these two ways of calculating the results is the inclusion of the SMT dataset of the 2013 task. When computing the results using Pearson's r, this dataset is included, but when computing the results using Spearman's ρ, it is not included.

For cross-lingual semantic similarity and semantic similarity in non-English languages, we evaluate on the STS tasks from SemEval 2017. This evaluation contains Arabic-Arabic, Arabic-English, Spanish-Spanish, Spanish-English, and Turkish-English datasets. The datasets were created by translating one or both pairs of an English STS pair into Arabic (`ar`), Spanish (`es`), or Turkish (`tr`). Following convention, we report results with Pearson's r for all systems, but also include results in Spearman's ρ in Table 10.

B.2 Question Retrieval

For our question retrieval evaluation, we report the accuracy (R@1) on the test sets of Natural Questions (NQ) (Kwiatkowski et al., 2019) shown in Table 11 and the Multilingual Knowledge Questions and Answers (MKQA) (Longpre et al., 2021) shown in Table 12. We use the the Probably Asked Questions dataset (PAQ) (Lewis et al., 2021) as a knowledge base from which we look up the nearest neighbor of each question in the NQ and MKQA test sets using cosine similarity.

B.3 Bitext Mining

For bitext mining, we use the Tatoeba dataset introduced in Artetxe and Schwenk (2019b) and the 2018 Building and Using Parallel Corpora (BUCC) shared bitext mining task (Zweigenbaum et al., 2018).

The Tatoeba dataset consists of 100-1000 pairs

Model	English Semantic Similarity				
	2012	2013	2014	2015	2016
CONTRASTIVE	69.7	61.1	76.4	81.4	77.7
BITRANSLATION	69.1	63.6	76.4	81.0	79.9
VMSST CONTRASTIVE	70.2	61.6	**76.5**	81.3	77.5
VMSST	**70.5**	**64.3**	**76.5**	**81.6**	**80.1**
CONTRASTIVE	68.0	74.9	69.1	79.9	76.9
BITRANSLATION	70.7	**77.9**	72.2	81.8	**79.7**
VMSST CONTRASTIVE	68.4	75.1	69.2	80.2	76.8
VMSST	**72.7**	**77.9**	**72.7**	**82.1**	79.2

Table 9: Full results on English STS. In the first part of the table, we show results, measured in Pearson's $r \times 100$, for each year of the STS tasks 2012-2016 as well as the average performance across all years. In the second part, we evaluate based on the Spearman's $\rho \times 100$ of the concatenation of the datasets of each year with the 2013 SMT dataset removed following (Reimers and Gurevych, 2019).

Model	Cross-Lingual Semantic Similarity									
	ar-ar		ar-en		es-es		es-en		tr-en	
CONTRASTIVE	72.4	72.2	72.7	74.2	79.7	81.0	71.7	72.0	77.2	77.0
BITRANSLATION	75.6	76.0	77.0	78.6	84.0	84.8	76.2	77.2	77.3	77.5
VMSST CONTRASTIVE	73.2	73.1	73.5	75.1	80.2	81.3	72.4	72.4	**77.8**	**78.0**
VMSST	**77.6**	**78.1**	**78.5**	**78.8**	**85.5**	**85.7**	**77.0**	**77.4**	77.0	77.0

Table 10: Full results on Cross-Lingual STS. We report results using both Pearson's $r \times 100$ and Spearman's $\rho \times 100$ across datasets, where Pearson's $r \times 100$ is the first column for each language pair and Spearman's $\rho \times 100$ is the second column.

Model	NQ
CONTRASTIVE	40.2
BITRANSLATION	**40.9**
VMSST CONTRASTIVE	40.3
VMSST	40.8

Table 11: Full results on question retrieval on the NQ data. We evaluate retrieval accuracy $\times 100$ using PAQ as a question knowledge base.

of data aligned to English for 112 languages. The accuracy for Tatoeba can be computed in two ways, depending if English is the target language or source language. We compute accuracy using cosine similarity in both directions for all 112 languages (19 are unseen in the training data) and average this score for all languages.

The goal of the BUCC task is to find the gold *aligned* parallel sentences given two corpora (one being very large) in two distinct languages. Languages are aligned with English and consist of German (de), French (fr), Russian (ru), and Chinese (zh). Following Schwenk (2018), we evaluate on the publicly available BUCC data. This involves scoring all pairs between the source target sentences and finding the optimal threshold that separates the data. Using the threshold, we can compute the precision, recall, and F_1 of the alignments. We report $F_1 \times 100$ in our results.

We compare two different approaches for finding the sentence alignments. In the first, BUCC (cosine), we compute the cosine similarity between the non-English source sentences and the English target sentences, selecting the highest scoring English sentence as the match. In the second, BUCC (margin), we follow Artetxe and Schwenk (2019a) and use a margin-based scoring approach.

C Full Training Data

We follow Artetxe and Schwenk (2019b) in constructing our training data, sampling data from Europarl,[14], United Nations (Rafalovitch and Dale, 2009),[15] OpenSubtitles2018 (Lison et al., 2018),[16], Global Voices,[17] Tanzil,[18] and Tatoeba v2021-07-

[14] http://opus.nlpl.eu/Europarl.php
[15] https://opus.nlpl.eu/UN.php
[16] http://opus.nlpl.eu/OpenSubtitles.php
[17] https://opus.nlpl.eu/GlobalVoices.php
[18] https://opus.nlpl.eu/Tanzil.php

22.[19]

The only deviation from their data sampling approach is that we take care to not include any Tatoeba test data in our training data. Our final corpus has nearly 216 million training examples, slightly less than 220 million reported in Artetxe and Schwenk (2019b). We use both English and Spanish as pivot languages, so each pair includes at least one English or Spanish sentence, and attempt to use approximately the same amount of data for each language if possible. We note that we only have training data for 92 languages instead of the 93 in Artetxe and Schwenk (2019b) due to not having training data for Aymara (ay). The full amount of English and Spanish parallel data used for each of the 92 languages is reported in Table 15.

[19] https://opus.nlpl.eu/Tatoeba.php

Model	MKQA											
Language	ar	da	de	en	es	fi	fr	he	hu	it	ja	km
CONTRASTIVE	21.4	30.4	29.2	33.2	30.4	27.7	30.0	24.4	26.9	29.4	24.2	23.6
BITRANSLATION	19.4	30.8	30.5	29.8	28.7	29.7	28.0	30.3	27.5	27.9	26.2	23.4
VMSST CONTRASTIVE	**24.6**	32.0	30.5	**33.4**	**31.6**	29.8	**30.9**	27.9	28.9	**31.0**	27.3	24.9
VMSST	21.4	**32.1**	**32.5**	31.5	30.3	**31.3**	30.0	**31.9**	**29.9**	30.0	**29.9**	**25.9**
Language	ko	ms	nl	no	pl	pt	ru	sv	th	tr	vi	zh
CONTRASTIVE	22.0	30.5	29.4	33.2	30.4	27.7	30.0	24.8	27.5	29.6	24.6	24.2
BITRANSLATION	19.4	30.8	30.9	30.0	29.2	30.2	28.1	30.4	28.0	28.3	26.9	23.7
VMSST CONTRASTIVE	**25.3**	32.3	30.6	**33.4**	**31.5**	30.1	**31.0**	28.1	29.9	**31.1**	27.7	24.9
VMSST	22.0	**32.4**	**32.8**	31.6	31.0	**31.6**	30.4	**32.2**	**30.4**	30.2	**30.4**	**26.2**

Table 12: Full results on question retrieval on the MKQA data. We evaluate retrieval accuracy $\times 100$ using PAQ as a question knowledge base.

Model	Cosine				Margin			
	de	fr	ru	zh	de	fr	ru	zh
CONTRASTIVE	84.6	81.3	66.6	64.4	**96.2**	**93.7**	**92.1**	**93.0**
BITRANSLATION	90.1	85.5	84.1	84.1	93.6	90.3	91.3	92.4
VMSST CONTRASTIVE	84.8	81.9	67.4	64.4	96.1	93.6	**92.1**	92.9
VMSST	**91.5**	**86.8**	**86.7**	**86.1**	94.3	91.0	91.8	92.8

Table 13: Full results on BUCC. We report results using both cosine similarity and the margin approach from (Artetxe and Schwenk, 2019a). Results are reported as $F_1 \times 100$.

Language	afr	amh	ang	ara	arq	arz	ast	awa	aze	bel	ben	ber	bos	bre	bul	cat
CONTRASTIVE	**97.6**	**94.9**	66.8	**95.0**	61.8	**86.1**	**91.3**	74.0	**95.8**	96.8	92.4	80.4	**97.5**	47.2	**96.4**	**97.8**
BITRANSLATION	94.8	84.2	42.9	94.0	42.0	80.3	80.3	56.3	91.3	95.0	91.0	72.6	96.8	18.9	95.3	96.8
VMSST CONTRASTIVE	97.4	93.5	**70.5**	94.7	**64.2**	85.7	90.9	**76.0**	**95.8**	**97.2**	**92.8**	**81.6**	97.3	**47.9**	96.2	**97.8**
VMSST	95.6	88.1	53.4	94.6	50.2	84.7	86.2	64.3	93.5	95.6	91.9	79.0	97.0	26.1	95.8	97.0

Language	cbk	ceb	ces	cha	cmn	cor	csb	cym	dan	deu	dsb	dtp	ell	epo	est	eus
CONTRASTIVE	86.1	62.3	**98.3**	**44.9**	97.5	35.1	67.8	**57.6**	**97.3**	**99.6**	75.4	18.9	**97.4**	**98.6**	98.6	96.9
BITRANSLATION	78.0	48.8	97.3	33.6	95.6	18.4	48.6	37.0	96.0	99.3	53.1	8.3	95.8	98.3	97.8	94.8
VMSST CONTRASTIVE	**86.9**	**63.7**	98.2	44.2	**97.7**	**37.6**	**69.8**	56.4	97.2	**99.6**	**76.9**	**20.3**	97.2	98.4	**98.8**	**97.1**
VMSST	83.7	52.8	97.9	38.0	96.4	23.3	56.1	43.0	96.8	99.2	63.2	10.2	97.0	98.2	98.2	95.4

Language	fao	fin	fra	fry	gla	gle	glg	gsw	heb	hin	hrv	hsb	hun	hye	ido	ile
CONTRASTIVE	90.3	98.0	96.4	**88.2**	**58.4**	**80.4**	**98.6**	52.1	**93.8**	**98.1**	**98.5**	80.1	**98.4**	**96.2**	93.0	92.8
BITRANSLATION	78.2	97.8	95.8	80.1	35.6	60.6	96.8	44.9	93.0	96.4	96.8	58.6	96.4	95.2	87.9	86.7
VMSST CONTRASTIVE	**91.0**	**98.2**	**96.6**	87.9	55.7	79.5	**98.6**	**53.8**	93.7	98.0	98.4	**82.4**	98.2	96.1	**94.4**	**93.1**
VMSST	82.6	98.0	96.0	83.5	39.4	62.7	97.4	50.4	**93.8**	97.5	97.5	68.9	96.8	94.7	91.9	90.2

Language	ina	ind	isl	ita	jav	jpn	kab	kat	kaz	khm	kor	kur	kzj	lat	lfn	lit
CONTRASTIVE	96.8	**97.0**	97.0	96.6	74.1	98.3	71.7	95.6	**92.8**	**87.8**	95.2	**76.3**	17.4	**89.9**	83.6	**98.2**
BITRANSLATION	94.9	95.2	96.3	96.2	62.9	96.8	60.8	93.9	86.1	85.4	92.5	60.5	8.8	83.9	74.6	97.5
VMSST CONTRASTIVE	**97.2**	96.9	**97.1**	96.6	**76.1**	**98.6**	**73.3**	**96.2**	92.3	87.5	**95.8**	76.0	**17.8**	89.8	**84.1**	98.1
VMSST	96.4	95.8	96.8	**96.8**	69.5	97.2	67.3	95.8	87.7	86.3	93.5	67.7	11.7	86.5	79.0	97.8

Language	lvs	mal	mar	max	mhr	mkd	mon	nds	nld	nno	nob	nov	oci	orv	pam	pes
CONTRASTIVE	98.0	**98.5**	94.5	**73.1**	**30.8**	97.6	94.4	90.5	**98.1**	96.5	**98.5**	**80.5**	77.9	66.5	14.0	95.5
BITRANSLATION	97.0	98.2	**95.0**	58.1	22.1	96.0	85.8	80.7	96.7	92.3	97.4	70.6	66.5	47.5	8.5	93.0
VMSST CONTRASTIVE	**98.1**	**98.5**	94.6	72.5	29.5	**98.0**	**95.0**	**92.1**	98.0	**97.0**	98.3	**80.5**	**78.0**	**67.4**	**14.6**	**95.7**
VMSST	97.5	98.3	**95.0**	63.6	26.8	96.4	89.8	84.5	97.3	93.5	97.6	76.3	72.1	55.9	9.9	94.5

Language	pms	pol	por	ron	rus	slk	slv	spa	sqi	srp	swe	swg	swh	tam	tat	tel
CONTRASTIVE	74.3	**99.0**	95.9	98.0	95.3	98.0	**97.0**	99.1	**98.6**	**96.6**	**97.5**	**76.8**	77.4	92.7	**92.0**	**98.1**
BITRANSLATION	62.1	97.2	95.7	97.6	95.0	97.4	96.3	98.8	98.1	95.6	96.8	48.7	67.1	90.9	82.8	96.2
VMSST CONTRASTIVE	**76.9**	98.9	**96.1**	**98.1**	**95.5**	**98.2**	**97.0**	**99.2**	**98.6**	96.5	**97.5**	73.2	**77.6**	**92.8**	**92.0**	97.6
VMSST	69.6	98.2	95.8	97.6	94.8	97.8	96.9	98.6	98.2	95.8	97.3	59.8	69.2	**92.8**	86.2	97.4

Language	tgl	tha	tuk	tur	tzl	uig	ukr	urd	uzb	vie	war	wuu	xho	yid	yue	zsm
CONTRASTIVE	96.0	97.8	44.6	**98.9**	**66.3**	76.1	96.0	**95.4**	**78.9**	98.2	54.0	93.5	74.6	92.3	**94.1**	**98.0**
BITRANSLATION	91.7	97.0	30.3	98.2	43.8	54.4	95.2	91.8	64.3	97.4	33.3	88.7	59.5	82.8	90.8	96.0
VMSST CONTRASTIVE	**96.4**	**98.2**	**47.5**	**98.9**	64.4	**77.6**	**96.3**	**95.4**	78.3	**98.4**	**54.5**	**93.8**	**75.7**	**92.6**	**94.1**	97.8
VMSST	93.0	97.5	38.7	98.8	56.7	64.1	95.5	93.5	68.7	97.9	37.7	91.0	63.7	86.0	93.0	96.6

Table 14: Full results on Tatoeba. We report results as accuracy $\times 100$.

Language	af	am	ar	ay	az	be	ber	bg
Training Pairs	77,772	101,613	7,907,914	0	291,925	6,330	142,061	4,834,661
Language	bn	br	bs	ca	cbk	cs	da	de
Training Pairs	1,148,461	34,472	4,166,739	895,940	1,623	5,429,060	7,767,119	8,707,293
Language	dtp	dv	el	en	eo	es	et	eu
Training Pairs	1,064	98,320	6,601,989	4,913,379	447,622	4,913,379	5,093,003	1,432,979
Language	fi	fr	ga	gl	ha	he	hi	hr
Training Pairs	7,785,493	8,935,842	1,112	391,824	134,775	4,046,554	358,907	3,911,368
Language	hu	hy	ia	id	ie	io	is	it
Training Pairs	5,256,214	8,194	12,048	4,326,151	2,445	3,181	2,712,556	8,468,538
Language	ja	ka	kab	kk	km	ko	ku	kw
Training Pairs	3,981,886	360,136	26,460	6,172	3,266	2,566,495	98,733	3,463
Language	kzj	la	lfn	lt	lv	mg	mhr	mk
Training Pairs	614	27,515	6,096	3,629,769	2,119,995	537,953	69	4,037,896
Language	ml	mr	ms	my	nb	nds	nl	oc
Training Pairs	867,026	52,340	3,288,492	4,802	9,694	6,263	8,346,102	730
Language	pl	ps	pt	ro	ru	sd	si	sk
Training Pairs	5,407,190	32	8,276,190	4,814,046	9,416,934	98,412	1,016,660	5,094,752
Language	sl	so	sq	sr	sv	sw	ta	te
Training Pairs	5,099,577	98,976	3,619,914	3,977,191	7,680,683	201,379	150,023	42,877
Language	tg	th	tl	tr	tt	ug	uk	ur
Training Pairs	135,245	3,849,777	34,829	5,854,059	132,273	101,989	1,687,685	844,052
Language	uz	vi	wuu	yue	zh			
Training Pairs	148,860	3,905,401	929	4,525	7,636,488			

Table 15: Full training data for each language. The total number of pairs is the sum of using English and Spanish as pivot languages.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In the Limitations section 8.

☒ A2. Did you discuss any potential risks of your work?
This is textual similarity model that does not generate text, trained on publicly available bitext. It does not carry any novel risks from previous generic textual embedding models.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Introduction is Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Yes, Appendix H.

☑ B1. Did you cite the creators of artifacts you used?
Yes, Appendix H.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
They all have licenses applicable to academic publishing and model release.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No need for discussion as our use (academic research) is consistent with the implied intended use of these datasets.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
These are well-studied standard academic translation datasets - there is no PII info and little offensitve content (possibly in the subtitles data for the latter).

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix H

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 6.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 6.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 6 and Section 8.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.