Learning Probabilistic Sentence Representations from Paraphrases

Mingda Chen **Kevin Gimpel**
Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA
{mchen,kgimpel}@ttic.edu

Abstract

Probabilistic word embeddings have shown effectiveness in capturing notions of generality and entailment, but there is very little work on doing the analogous type of investigation for sentences. In this paper we define probabilistic models that produce distributions for sentences. Our best-performing model treats each word as a linear transformation operator applied to a multivariate Gaussian distribution. We train our models on paraphrases and demonstrate that they naturally capture sentence specificity. While our proposed model achieves the best performance overall, we also show that specificity is represented by simpler architectures via the norm of the sentence vectors. Qualitative analysis shows that our probabilistic model captures sentential entailment and provides ways to analyze the specificity and preciseness of individual words.

1 Introduction

Probabilistic word embeddings have been shown to be useful for capturing notions of generality and entailment (Vilnis and McCallum, 2014; Athiwaratkun and Wilson, 2017; Athiwaratkun et al., 2018). In particular, researchers have found that the entropy of a word roughly encodes its generality, even though there is no training signal explicitly targeting this effect. For example, hypernyms tend to have larger variance than their corresponding hyponyms (Vilnis and McCallum, 2014). However, there is very little work on doing the analogous type of investigation for sentences.

In this paper, we define probabilistic models that produce distributions for sentences. In particular, we choose a simple and interpretable probabilistic model that treats each word as an operator that translates and scales a Gaussian random variable representing the sentence. Our models are able to capture sentence specificity as measured by the annotated datasets of Li and Nenkova (2015) and Ko et al. (2019) by training solely on noisy paraphrase pairs. While our "word-operator" model yields the strongest performance, we also show that specificity is represented by simpler architectures via the norm of the sentence vectors. Qualitative analysis shows that our models represent sentences in ways that correspond to the entailment relationship and that individual word parameters can be analyzed to find words with varied and precise meanings.

2 Proposed Methods

We propose a model that uses ideas from flow-based variational autoencoders (VAEs) (Rezende and Mohamed, 2015; Kingma et al., 2016) by treating each word as an "operator". Intuitively, we assume there is a random variable z associated with each sentence $s = \{w_1, w_2, \cdots, w_n\}$. The random variable initially follows a standard multivariate Gaussian distribution. Then, each word in the sentence transforms the random variable sequentially, leading to a random variable that encodes its semantic information.

Our word linear operator model (WLO) has two types of parameters for each word w_i: a scaling factor $A_i \in \mathbb{R}^k$ and a translation factor $B_i \in \mathbb{R}^k$. The word operators produce a sequence of random variables $z_0, z_1, \cdots, z_n$ with $z_0 \sim \mathcal{N}(0, I_k)$, where I_k is a $k \times k$ identity matrix, and the operations are defined as

$$z_i = A_i(z_{i-1} + B_i) \tag{1}$$

The means and variances for each random variable are computed as follows:

$$\mu_i = A_i(\mu_{i-1} + B_i) \tag{2}$$

$$\Sigma_i = A_i \Sigma_{i-1} A_i^\top \tag{3}$$

For computational efficiency, we only consider diagonal covariance matrices, so the equations above can be further simplified.

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 17–23
July 9, 2020.

3 Learning

Following Wieting and Gimpel (2018), all of our models are trained with a margin-based loss on paraphrase pairs (s_1, s_2):

$$\max(0, \delta - d(s_1, s_2) + d(s_1, n_1)) + \\ \max(0, \delta - d(s_1, s_2) + d(s_2, n_2))$$

where δ is the margin and d is a similarity function that takes a pair of sentences and outputs a scalar denoting their similarity. The similarity function is maximized over a subset of examples (typically, the mini-batch) to choose negative examples n_1 and n_2. When doing so, we use "mega-batching" (Wieting and Gimpel, 2018) and fix the mega-batch size at 20. For deterministic models, d is cosine similarity, while for probabilistic models, we use the expected inner product of Gaussians.

3.1 Expected Inner Product of Gaussians

Let μ_1, μ_2 be mean vectors and Σ_1, Σ_2 be the variances predicted by models for a pair of input sentences. For the choice of d, following Vilnis and McCallum (2014), we use the expected inner product of Gaussian distributions:

$$\begin{aligned}
&\int_{x \in \mathbb{R}^k} \mathcal{N}(x; \mu_1, \Sigma_1)\mathcal{N}(x; \mu_2, \Sigma_2) dx \\
&= \log \mathcal{N}(0; \mu_1 - \mu_2, \Sigma_1 + \Sigma_2) \\
&= -\frac{1}{2} \log\det(\Sigma_1 + \Sigma_2) - \frac{d}{2}\log(2\pi) \\
&\quad - \frac{1}{2}(\mu_1 - \mu_2)^\top (\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)
\end{aligned} \tag{4}$$

For diagonal matrices Σ_1 and Σ_2, the equation above can be computed analytically.

3.2 Regularization

To avoid the mean or variance of the Gaussian distributions from becoming unbounded during training, resulting in degenerate solutions, we impose prior constraints on the operators introduced above. We force the transformed distribution after each operator to be relatively close to $\mathcal{N}(0, I_k)$, which can be thought of as our "prior" knowledge of the operator. Then our training additionally minimizes

$$\lambda \sum_{s \in \{s_1, s_2, n_1, n_2\}} \sum_{w \in s} KL(\mathcal{N}(\mu(w), \Sigma(w)) \| \mathcal{N}(0, I))$$

where λ is a hyperparameter tuned based on the performance on the 2017 semantic textual similarity (STS; Cer et al., 2017) data. We found prior regularization very important, as will be shown in our results. For fair comparison, we also add L2 regularization to the baseline models.

Domain	News	Twitter	Yelp	Movie
Number of instances	900	984	845	920

Table 1: Sizes of test sets for sentence specificity.

4 Experiments

4.1 Baseline Methods

We consider two baselines that have shown strong results on sentence similarity tasks (Wieting and Gimpel, 2018). The first, word averaging (WordAvg), simply averages the word embeddings in the sentence. The second, long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997) averaging (LSTMAvg), uses an LSTM to encode the sentence and averages the hidden vectors. Inspired by sentence VAEs (Bowman et al., 2016), we consider an LSTM based probabilistic baseline (LSTMGaussian) which builds upon LSTMAvg and uses separate linear transformations on the averaged hidden states to produce the mean and variance of a Gaussian distribution.

We also benchmark several pretrained models, including GloVe (Pennington et al., 2014), Skip-thought (Kiros et al., 2015), InferSent (Conneau et al., 2017), BERT (Devlin et al., 2019), and ELMo (Peters et al., 2018). When using GloVe, we either sum embeddings (GloVe SUM) or average them (GloVe AVG) to produce a sentence vector. Similarly, for ELMo, we either sum the outputs from the last layer (ELMo SUM) or average them (ELMo AVG). For BERT, we take the representation for the "[CLS]" token.

4.2 Datasets

We use the preprocessed version of ParaNMT-50M (Wieting and Gimpel, 2018) as our training set, which consists of 5 million paraphrase pairs.

For evaluating sentence specificity, we use human-annotated test sets from four domains, including news, Twitter, Yelp reviews, and movie reviews, from Li and Nenkova (2015) and Ko et al. (2019). For the news dataset, labels are either "general" or "specific" and there is additionally a training set. For the other datasets, labels are real values indicating specificity. Statistics for these datasets are shown in Table 1.

For analysis we also use the semantic textual similarity (STS) benchmark test set (Cer et al., 2017) and the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015).

4.3 Specificity Prediction Setup

For predicting specificity in the news domain, we threshold the predictions either based on the entropy of Gaussian distributions produced from probabilistic models or based on the norm of vectors produced by deterministic models, which includes all of the pretrained models. The threshold is tuned based on the training set but no other training or tuning is done for this task with any of our models. For prediction in other domains, we simply compute the Spearman correlations between the entropy/norm and the labels.

Intuitively, when sentences are longer, they tend to be more specific. So, we report baselines ("Length") that predict specificity solely based on length, by thresholding the sentence length for news (choosing the threshold using the training set) or simply returning the length for the others. The latter results are reported from Ko et al. (2019). We also consider baselines that average or sum ranks of word frequencies within a sentence ("Word Freq. AVG" and "Word Freq. SUM").

5 Results

5.1 Sentence Specificity

Table 2 shows results on sentence specificity tasks. We compare to the best-performing models reported by Li and Nenkova (2015) and Ko et al. (2019). Their models are specifically designed for predicting sentence specificity and they both use labeled training data from the news domain.

Our averaging-based models (WORDAVG, LSTMAVG) failed on this task, either giving the majority class accuracy or negative correlations. So, we also evaluate WORDSUM, which sums word embeddings instead of averaging and shows strong performance compared to the other models.

While the model from Li and Nenkova (2015) performs quite well in the news domain, its performance drops on other domains, indicating some amount of overfitting. On the other hand, WORDSUM and WLO, which are trained on a large number of paraphrases, perform consistently across the four domains and both outperform the supervised models on Yelp. Additionally, our WLO model outperforms all our other models, achieving comparable performance to the supervised methods.

	News	Twitter	Yelp	Movie
Majority baseline	54.6	-	-	-
Length	73.4	44.5	67.6	58.1
Word Freq. SUM	55.5	10.1	54.6	22.1
Word Freq. AVG	61.5	0.0	28.5	0.0
Prior work trained on labeled sentence specificity data				
Li and Nenkova (2015)	81.6	55.3	63.3	57.5
Ko et al. (2019)	-	67.9	75.0	70.6
Sentence embeddings from pretrained models				
GloVe SUM	70.4	32.2	62.8	49.0
GloVe AVG	54.6	-49.6	-59.0	-38.2
InferSent	75.0	60.5	76.6	61.2
Skip-thought	57.7	2.9	14.1	27.2
BERT	64.5	20.8	29.5	18.1
ELMo SUM	65.4	46.2	72.7	59.3
ELMo AVG	56.2	-9.4	-0.9	-22.5
Our work				
WORDAVG	54.6	-10.6	-32.3	-27.2
WORDSUM	75.8	57.9	75.4	60.0
LSTMAVG	54.6	-14.8	-41.1	-14.8
LSTMGAUSSIAN	55.5	3.2	2.2	4.1
WLO	77.4	60.5	76.6	61.9

Table 2: Sentence specificity results on test sets from four domains (accuracy (%) for News and Spearman correlations (%) for others). Highest numbers for the models described in this work are underlined.

	Full	Length norm.
Majority baseline	54.6	50.1
WORDAVG	54.6	69.0
WORDSUM	75.8	68.6
LSTMAVG	54.6	69.6
LSTMGAUSSIAN	55.5	67.0
WLO	**77.4**	**70.1**

Table 3: Accuracy (%) for the specificity News test set, in both the original and length normalized conditions. Highest numbers in each column are in bold.

Among pretrained models, BERT, Skip-thought, ELMo SUM, and GloVe SUM show slight correlations with specificity, while InferSent performs strongly across domains. InferSent uses supervised training on a large manually-annotated dataset (SNLI) while WORDSUM and WLO are trained on automatically-generated paraphrases and still show results comparable to InferSent.

To control for effects due to sentence length, we design another experiment in which sentences from News training and test are grouped by length, and thresholds are tuned on the group of length k and tested on the group of length $k-1$, for all k, leading to a pool of 3582 test sentences.

Table 3 shows the results. In this length-normalized experiment, the averaging models demonstrate much better performance and even outperform WORDSUM, but still WLO has the best performance.

	Entailment	Neutral	Contradiction
GloVe	42.5	53.8	39.6
InferSent	78.3	57.2	55.7
Skip-thought	62.5	54.3	57.3
ELMo	78.3	58.3	63.4
BERT	65.0	55.7	56.3
WORDAVG	77.5	50.0	57.2
WORDSUM	75.0	54.7	57.7
LSTMAVG	71.7	49.5	52.4
LSTMGAUSSIAN	65.0	49.5	48.6
WLO	75.8	54.7	57.2

Table 4: Percentage of cases in which hypothesis has larger entropy (or smaller norm for non-probabilistic models) than premise for equal-length sentence pairs in the SNLI test set. In this setting, GloVe and ELMo would give the same results under either SUM or AVG.

6 Analysis

6.1 Sentence Entailment

Vilnis and McCallum (2014) explored whether their Gaussian word entropies captured the lexical entailment relationship. Here we analyze the extent to which our representations capture sentential entailment.

We test models on the SNLI test set, assuming that for a given premise p and hypothesis h, p is more specific than h for entailing sentence pairs. To avoid effects due to sentence length, we only consider $\langle p, h \rangle$ pairs with the same length. After this filtering, entailment/neural/contradiction categories have 120/192/208 instances respectively. We encode each sentence and calculate the percentage of cases in which the hypothesis has larger entropy (or smaller norm for non-probabilistic models) than the premise. Under an ideal model, this would happen with 100% of entailing pairs while showing random results (50%) for the other two types of pairs.

As shown in Table 4, our best paraphrase-trained models show similar trends to InferSent, achieving around 75% accuracy in the entailment category and around 50% accuracy in other categories. Although ELMo can also achieve similar accuracy in the entailment category, it seems to conflate entailment with contradiction, where it shows the highest percentage of all models. Other models, including BERT, GloVe, and Skip-thought, are much closer to random (50%) for entailing pairs.

6.2 Lexical Analysis

WLO associates translation and scaling parameters with each word, allowing us to analyze the impact of words on sentence representations. We ranked words under several criteria based on their translation parameter norms and single-word sentence entropies. Table 5 shows the top 20 words under each criterion.

Small norm		Large norm	
small abs. ent.	small ent.	small abs. ent.	small ent.
,	addressing	staveb	cenelec
/	derived	jerusalem	ohim
by	decree	trent	placebo
an	fundamental	microwave	hydrocarbons
gon	beneficiaries	brussels	iec
as	tendency	synthetic	paras
having	detect	christians	allah
a	reservations	elephants	milan
on	remedy	seldon	madrid
for	eligibility	burger	±
from	film-coated	experimental	ukraine
’d	breach	alison	intravenous
—	exceed	63	electromagnetic
his	flashing	prophet	131
’	objectives	diego	electrons
upon	cue	mallory	northeast
under	commonly	ö	blister
towards	howling	natalie	http
’s	vegetable	hornblower	renal
with	bursting	korea	asteroid

Table 5: Examples showing top-20 lists of large-norm or small-norm words ranked based on small absolute entropy or small entropy in WLO.

Words with small norm and small absolute entropy have little effect, both in terms of meaning and specificity; they are mostly function words. Words with large norm and small entropy have a large impact on the sentence while also making it more specific. They are organization names (*cenelec*) or technical terms found in medical or scientific literature. When they appear in a sentence, they are very likely to appear in its paraphrase.

Words with large norm and small absolute entropy contribute to the sentence semantics but do not make it more specific. Words like *microwave* and *synthetic* appear in many contexts and have multiple senses. Names (*trent*, *alison*) also appear in many contexts. Words like these often appear in a sentence's paraphrase, but can also appear in many other sentences in different contexts.

Words with small norm/entropy make sentences more specific but do not lend themselves to a precise characterization. They affect sentence meaning, but can be expressed in many ways. For example, when *beneficiaries* appears in a sentence, its paraphrase often has a synonym like *beneficiary*, *heirs*, or *grantees*. These words may have multiple senses, but it appears more that they correspond to

WORDSUM		WLO	
largest norm (specific)	smallest norm (general)	smallest entropy (specific)	largest entropy (general)
this regulation shall not apply to wine grape products, with the exception of wine vinegar, spirit drinks or flavoured wines.	oh, man, you're gonna... you're just gonna get it, vause*, aren't you ?	under a light coating of dew she was a velvet study in reflected mauve with rose overtones against the indigo nightward* sky.	oh, man, you're gonna... you're just gonna get it, vause*, aren't you?
operating revenue community subsidies other subsidies/revenue* total (a) operating expenditure staff administration operating activities total (b) operating result (c=ab)	okay, i know you don't get relationships, like, at all, but i don't need to screw anyone for an "a."	a similar influenza disease occurred in 47% of patients who received plegridy 125 micrograms every 2 weeks, and 13% of the patients were given placebo.	'authorisation' means an instrument issued in any form by the authorities by which the right to carry on the business of a credit institution is granted;

Table 6: Examples of most general and specific sentences for selected lengths (* = mapped to unknown symbol).

	With Prior		Without Prior	
	Acc.	F_1	Acc.	F_1
WLO	77.4	78.4	67.9	68.2

Table 7: Accuracy (%) and F_1 score (%) for specificity News test set with and without prior regularization.

	STS Benchmark
WORDAVG	73.4
LSTMAVG	73.6
LSTMGAUSSIAN	**74.3**
WLO	73.7

Table 8: Pearson correlation (%) for STS benchmark test set. Highest number is in bold.

concepts with many valid ways of expression.

6.3 Sentential Analysis

We subsample the ParaNMT training set and group sentences by length. For each model and length, we pick the sentence with either highest/lowest entropy or largest/smallest norm values. Table 6 shows some examples. WORDSUM tends to choose conversational sentences as general and those with many rare words as specific. WLO favors literary and technical/scientific sentences as most specific, and bureaucratic/official language as most general.

6.4 Effect of Prior Regularization

As shown in Table 7, there is a large performance improvement after adding prior regularization for avoiding degenerate solutions.

6.5 Semantic Textual Similarity

Although semantic textual similarity is not our target task, we still include the performance of our models on the STS benchmark test set in Table 8 to show that our models are competitive with standard strong baselines. When using probabilistic models to predict sentence similarity during test time, we let $v_1 = concat(\mu_1, \Sigma_1)$, $v_2 = concat(\mu_2, \Sigma_2)$, where *concat* is a concatenation operation, and predict sentence similarity via $cosine(v_1, v_2)$, since we find it performs better than solely using the mean vectors. The two probabilistic models, LSTMGAUSSIAN and WLO, are able to outperform the baselines slightly.

7 Related Work

Our models are related to work in learning probabilistic word embeddings (Vilnis and McCallum, 2014; Athiwaratkun and Wilson, 2017; Athiwaratkun et al., 2018) and text-based VAEs (Miao et al., 2016; Bowman et al., 2016; Yang et al., 2017; Kim et al., 2018; Xu and Durrett, 2018, *inter alia*). The WLO is also related to flow-based VAEs (Rezende and Mohamed, 2015; Kingma et al., 2016), where hidden layers are viewed as operators over the density function of latent variables.

Previous work on sentence specificity relies on hand-crafted features or direct training on annotated data (Louis and Nenkova, 2011; Li and Nenkova, 2015). Recently, Ko et al. (2019) used domain adaptation for this problem when only the source domain has annotations. Our work also relates to learning sentence embeddings from paraphrase pairs (Wieting et al., 2016; Wieting and Gimpel, 2018).

8 Conclusion

We trained sentence models on paraphrase pairs and showed that they naturally capture specificity and entailment. Our proposed WLO model, which treats each word as a linear transformation operator, achieves the best performance and lends itself to analysis.

Acknowledgments

We would like to thank the anonymous reviewers, NVIDIA for donating GPUs used in this research, Jessy Li for clarifying the experimental setup used in Li and Nenkova (2015), and Google for a faculty research award to K. Gimpel that partially supported this research.

References

Ben Athiwaratkun and Andrew Wilson. 2017. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1645–1656. Association for Computational Linguistics.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Melbourne, Australia. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-amortized variational autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2678–2687, Stockholmsmssan, Stockholm Sweden. PMLR.

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

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3294–3302, Cambridge, MA, USA. MIT Press.

Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2019. Domain agnostic real-valued specificity prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6610–6617. AAAI Press.

Junyi Jessy Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2281–2287. AAAI Press.

Annie Louis and Ani Nenkova. 2011. Automatic identification of general and specific sentences by leveraging discourse annotations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 605–613. Asian Federation of Natural Language Processing.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1727–1736. JMLR.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Luke Vilnis and Andrew McCallum. 2014. Word representations via Gaussian embedding. *arXiv preprint arXiv:1412.6623*.

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

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462. 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. 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 *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890, International Convention Centre, Sydney, Australia. PMLR.

A Supplementary Material

A.1 Hyperparameters

For all experiments, the dimension of word embeddings and word operator is 50. The dimension of LSTM is 100. The dimension of Gaussian distribution for LSTMGaussian is 100. Mini-batch size is 100. For LSTM, LSTMGaussian, and WLO, we scramble training sentences with a probability of 0.4. For baseline models, the margin δ is 0.4. For other models, δ is 1. All models are randomly initialized and trained with Adam (Kingma and Ba, 2014) using learning rate of 0.001.