Paper 7578: Paraphrase Generation with Latent Bag of Words

We thank all reviewers for their detailed constructive feedback and suggestions.

Major concerns/clarifications:

- **Clarifying a critical error:** First, we have noticed that both Reviewer 1 and Reviewer 2 suggest that the latent BOW is merely taking an average representation of the bag of words as the decoder initial state. We emphasize that this is **not correct**, and we apologize for our paper leading to this misunderstanding. Critically, the decoder also performs attention to the BOW (appendix line 42-43, source code codes/src/latent_bow.py line 207), precisely as requested by Reviewer 1. We will clarify this in the paper.
- **New results/model enhancements further to Reviewer 1's and Reviewer 3's main concern**: We agree that a "more complex generative process" would enhance the paper. Accordingly, to better exploit the BOW information, we now condition the decoder's inputs on the mixed BOW embeddings (with z_{ij}), and further integrate the *Copy Mechanism* (Gu et al., 2016; See et al., 2017), directly copying a word from BOW as the output. These mechanisms all yield constant improvements (Table 4). We will also update the results section in the paper and release the code.

We think these two enhancements (and the clarity around our existing use of attention) improve the paper considerably, and we ask the reviewers to reconsider the contributions in light of this clarification and enhancement.

Additional important concerns:

- **Comparison with previous works (reviewer 1 and 3):** Thank you; this is an important point, and we have improved the paper considerably on this point. First, although we did not mention this explicitly, our baseline model, seq2seq-attn has basically *identical architecture* as the Residual LSTM (Prakash et al., 16). On the quora dataset, the SOTA model is RbM with inverse reinforcement learning (Li et al., 2018). Since they do not release the code, we list our implementation results and theirs reported on Table 1. Generally we have close numbers. Their model has better scores than ours, which may come from (a) they use twice the size of training set, (b) they directly optimize the BLEU and ROUGE scores. Our advantages are the model transparency and interpretability. On the MSCOCO dataset, the baseline model is Prakash et al (2016), but without released code. We are unsure about many details (train-test split, BLEU ngrams etc.). The experiments in our paper are on MSCOCO17, and Prakash et al (2016) is on MSCOCO14. So we redo our experiments on MSCOCO14 and try to make the settings as comparable as possible, with the results in Table 2. Generally we have comparable numbers. Also we will release all implementations in an effort to establish a fair comparison for future research.
- **More samples (reviewer 1 and 3):** The comparison between LBOW and seq2seq is listed in Table 3. Generally our model has better word choice because of the BOW. More samples from our model are in the appendix.
- **The paraphrase task itself (reviewer 3):** We view paraphrase generation as a reliable benchmark task since it also requires meaningful word choice and ordering, and hence it is our focus in this work. We agree other tasks like data-to-text are challenging and important, so on your recommendation we are now implementing the experiments on the Wikibio dataset(Lebret et. al. 16). Our preliminary results (table 5) have close numbers with SOTA model (Li et. al. 18) and indicate the value of this model in that task as well; we will complete the results for publication.

Table 1. Quora results comparison between ours and the SOTA (Li .et .al 18), dispite different implementations, the numbers are close

Li .et .al (18)	R2	B2	Our Implementation	R2	B2
Seq2seq	31.47	36.55	Seq2seq	33.04	40.41
Residual LSTM	32.43	37.38	Residual LSTM	32.86	40.49
RbM-SL	38.11	43.54	LBOW-topK	34.57	42.03
RbM-IRL	37.72	43.09	LBOW-gumbel	34.47	41.96

Table 2. MSCOCO 14 results compared with the baseline. The implementation details of the baseline model are unclear. If the bleu reported by Prakash .et.al (16) is bleu3, then we have close numbers

Prakash .et .al (16)	Bleu	Our implementation	Bleu3
seq2seq-attn (vanilla)	33.1	seq2seq-attn (vanilla)	33.94
seq2seq-attn (residual)	37.0	seq2seq-attn (residual)	33.96
-	-	LBOW	35.71
4 layer lstm, 512 hidden, 0.5 dropout, bleu ngram unspecified		4 layer lstm, 512 hidden, 0.5 dropout, bleu 3	

Table 3. Model ourputs comparison. Our model generally has a better word choice due to the predicted BOW.

Input	what are some ways to build your blog audience
S2S-Attn	how do i create a blog
LBOW	how do i build my blog audience
Input	can you name great works of art inspired by atheism
S2S-Attn	can you the art of mind
LBOW	can you name a great name of atheism
Input	is there somewhere i can host my django web app
S2S-Attn	can i host my app
LBOW	how can i host my web app
Input	what are the best ways to build up my credit score
S2S-Attn	what are some ways to build up with a credit
LBOW	how do i build up credit score

Table 4. Exploit the BOW information with different components. Adding more sophisticated techniques to the BOW yields consistent improvements

Quora	B1	B2	R1	R2
seq2seq	54.62	40.41	57.27	33.04
seq2seq-attn	54.59	40.49	57.1	32.86
LBOW	55.79	42.03	58.79	34.57
LBOW + BOW emb	56.16	42.14	58.66	34.36
LBOW + Copy	56.53	42.67	59.85	35.30

Table 5. prelimilary results on data to text genetaion. Our method shows improvements and has comparable numbers with SOTA

Our Implementation	B4	R2
Seq2seq-attn	40.82	52.48
LBOW + Copy	42.00	53.55
Liu et.al.(18)	B4	R2
Seq2seq-attn	43.65	-
Structure-aware S2S	44.89	-