

# Beam Search with Bidirectional Strategies for Neural Response Generation

Pierre Colombo<sup>†\*</sup>, Chouchang (Jack) Yang<sup>\*</sup>, Giovanna Varni<sup>\*</sup>, Chloé Clavel<sup>\*</sup>

<sup>\*</sup>Télécom ParisTech, Université Paris Saclay

<sup>†</sup> IBM GBS France

<sup>\*</sup> University of Washington

pierre.colombo@ibm.com

ccjack@uw.edu

giovanna.varni@telecom-paris.fr

chloe.clavel@telecom-paris.fr

## Abstract

Sequence-to-sequence neural networks have been widely used in language-based applications as they have flexible capabilities to learn various language models. However, when seeking for the optimal language response through trained neural networks, current existing approaches such as beam-search decoder strategies are still not able reaching to promising performances. Instead of developing various decoder strategies based on a “regular sentence order” neural network (a trained model by outputting sentences from left-to-right order), we leveraged “reverse” order as additional language model (a trained model by outputting sentences from right-to-left order) which can provide different perspectives for the path finding problems. In this paper, we propose bidirectional strategies in searching paths by combining two networks (left-to-right and right-to-left language models) making a bidirectional beam search possible. Besides, our solution allows us using any similarity measure in our sentence selection criterion. Our approaches demonstrate better performance compared to the unidirectional beam search strategy.

## 1 Introduction

Seq2seq models have shown state-of-the-art performance in tasks such as machine translation (Sutskever et al., 2014), neural conversation (Li et al., 2016b), image captioning (Venugopalan et al., 2015), and text summarization (Nallapati et al., 2016), exhibiting human-level performance (Johnson et al., 2017). Despite some drawbacks (e.g huge parallel corpora are needed to train seq2seq models, expert knowledge required to set the hyperparameters), seq2seq models are becoming increasingly popular and are now deployed in real world applications (McCann et al., 2019). During inference, a trained seq2seq model aims to find

the best sentence given a source sentence. Since searching for all the possible paths is not practical but also computationally expensive, existing work relies on beam search based algorithms to solve this issue (Lipton, 2015). Current solutions have limited performances due to three major constraints: (1) beam search sentence selection is based on likelihood regardless of the evaluation metric, (2) during the generation process, only left to right dependencies (right to left are ignored) are considered by the model, (3) seq2seq strongly foster safe sentences (Li et al., 2016a): during generation, the influence of the input decreases while words are generated (Zhang et al., 2018) meaning that the end of the sequence is less likely to be input relevant. Those limitations constraint beam search performances: on Switchboard Corpus with a Beam Size of 50 optimal re-ranking would yield to an improvement of 128% (see Supplementary for full table). For the evaluation metric we follow Li et al. (2016a); Colombo et al. (2019) adopt the BLEU-4 score to compare the algorithms.

In this work, we introduce two novel algorithms based on beam search (*VBS*) with different ranking procedure: (1) *BidiS* a generalisation of the work of Wen et al. (2015); Mimura et al. (2018) that uses a “reverse” decoder to re-score the produced sentence penalizing sentences whose end is less likely given the input, (2) *BidiA* an algorithm, that looks at the closest pair by incorporating the similarity measure between two beams of hypothesis; one with sentences generated in the regular order, the other with sentences generated in the reverse order. Our results show that leveraging the reverse order can boost beam search performance leading to higher BLEU-4 score and more diverse responses compared to *VBS*. Complexity analysis further shows that our proposed algorithms have dramatically reduced computational cost compared to the traditional approaches.

## 2 Models

We notice that limitation (2) and (3) previously introduced can be solved by introducing bidirectionally in the beam search. Indeed, training a seq2seq to output the sentence in the reverse order can model right to left dependencies and reduces the path length between the input and the end of the sentence (the shorter the paths are, the easier it is to model dependencies) making the end of the sentence more dependent of the input and producing more diverse sequences and model right to left dependencies as well.

### 2.1 Preliminaries

**Vanilla Beam Search (VBS)** We denote  $B$  as the beam size,  $T$  as the maximum sentence length and  $V$  as the vocabulary size. A RNN encoder takes an input sequence  $X = (x_1, \dots, x_T)$ , to learn the language model word by word during the training phase. When a language task is given, the decoder travels through paths at each time and keeps the  $B$  most likely sequences. At each step, *VBS* considers at most  $V \times B$  hypothesis. The sequence likelihood is measured by using the score function in (Wu et al., 2016)

$$s(Y, X) = \frac{\log P(Y|X)}{lp(Y)} \quad (1)$$

where  $X$  is the source,  $Y$  is the current target, and  $lp(Y) = \frac{(5+|Y|)^\alpha}{(5+1)^\alpha}$  is the length normalization factor. We select  $\alpha = 0.6$ , which produce a higher BLEU score as illustrated in (Wu et al., 2016). The beam search is stopped when exactly  $B$  finished candidates have been found (Luong et al., 2015). In the worst case, the algorithm will run for a maximum of  $T$  steps.

**Regular and Reverse model training** For the bidirectional beam search we train two different networks over the same dataset. The first seq2seq network called "regular" is trained to predict the sentence in the regular order. The second network called "reverse" is trained to predict the sentence in the reversed order. For example, if the regular network is trained with the pair "What do you like ?" / "I like cats !", the reversed is trained with the pair "What do you like ?" / "! cats like I". During decoding, the reverse model estimates right-to-left dependencies while the regular model estimates left-to-right dependencies.<sup>1</sup> Two different settings

<sup>1</sup>From graph topology viewpoints, the decoder procedure

have been explored: (1) training two independent seq2seq, (2) sharing the encoder of the two seq2seq and training using the following loss  $\mathcal{L}$  where:

$$\mathcal{L} = \alpha \mathcal{L}_{Rg} + (1 - \alpha) * \mathcal{L}_{Re} \quad (2)$$

$\mathcal{L}_{Rg}$  is the Cross Entropy loss computed with the regular decoder and  $\mathcal{L}_{Re}$  is the Cross Entropy loss computed with the reverse decoder. Since both approaches exhibit comparable performances, we choose to share the encoder to minimise the number of parameters in our model.

### 2.2 Beam Search with Bidirectional Scoring (*BidiS*)

A Beam search generates word by word from left to right: the token generated at time step  $t$  only depending on past token, but would not affected by the future tokens. Inspired by the work of (Li et al., 2016a), we propose a Beam Search with Bidirectional Scoring (*BidiS*), which scores the  $B$  best candidates generated by the regular seq2seq model as follows:

$$s(Y_T, X) = \frac{\log P(Y_T^+|X)}{lp(Y_T)} + \lambda \times \frac{\log P(Y_T^-|X)}{lp(Y_T)} \quad (3)$$

where  $Y_T^+$  and  $Y_T^-$  represents the final sequence in the regular order and reversed order respectively. Moreover,  $P(Y_T^+|X)$  is computed by using the regular model while  $P(Y_T^-|X)$  is computed by using the reverse model.  $\lambda^2$  is optimized in the validation. The intuition here is as follows: after generation of  $B$  sentences from the regular seq2seq, the reverse model computes  $P(Y_T^-|X)$ , and assigns higher probabilities to sequences presenting a more likely right-to-left structure and more likely ending given the input. Since  $B$  best lists produced by our models are grammatically correct, the final selected options are well-formed and present the best combination of both directions.

### 2.3 Beam Search with bidirectional agreement (*BidiA*)

The previous algorithm has two weaknesses. Firstly it introduces a hyperparameter  $\lambda$ . Secondly, the reverse model is only used to re-score the sentences

from the right side is very different from the left sides. During exploring graph from left to right the regular seq2seq faces a huge very likely first token while the reverse seq2seq has a very restraint choice (mainly punctuation). More details are included in Supplementary.

<sup>2</sup>Optimisation process shows that  $\lambda$  compensate the difference of scale between  $\frac{\log P(Y_T^+|X)}{lp(Y_T)}$  and  $\frac{\log P(Y_T^-|X)}{lp(Y_T)}$

generated by the regular model, meaning that potentially good sentences generated by the reverse model are not considered. We solve these two problems by proposing a Beam Search with bidirectional agreement (*BidiA*); a hyperparameter free algorithm that uses  $\frac{B}{2}$  best sequences according to the reverse seq2seq model. Formally if  $S$  and  $S'$  are the sets containing  $\frac{B}{2}$  sequence generated by the regular and reverse model respectively, we output  $Y_{n_0}$  such that:

$$(Y_{n_0}, Y_{r_0}) = \arg \min_{Y_n \in S, Y_r \in S'} 1 - \text{sim}(Y_n, Y_r) \quad (4)$$

where  $\text{sim}$  represents any similarity measure between two sentences<sup>3</sup>. For our experiment, we propose two different choices: (1) an adaptation of the BLEU score,  $BLEU_T$  where the corpus length is set to  $T$  to foster longer responses formally:

$$BLEU_T = BP_T \times \exp\left(\sum_{n=1}^N w_n \log(p_n)\right) \quad (5)$$

where the brevity penalty is set to  $BP_T = \exp(1 - \frac{T}{c})^4$ ,  $p_n$  is the geometric average of the modified n-gram precisions, using n-grams up to length  $N$  and  $w_n$  are positive weights summing to one, (2) an adaptation of the Word Mover’s Distance ( $WMD_T$ ) (Kusner et al., 2015) (stopwords are removed and final score is multiplied by  $BP_T$ ) that captures the relationship between words, by computing the “transportation” from one phrase to another conveyed by each word<sup>5</sup>.

### 3 Results

#### 3.1 Corpora and Metrics

**Corpora:** We evaluate our algorithms on two spoken datasets (specific phenomena appear when working with spoken language (Dinkar et al., 2020) compared to written text). (1) The Switchboard Dialogue Act Corpus (SwDA) a telephone speech corpus (Stolcke et al., 1998), consisting of about 2.400 two-sided telephone conversation. (2) The Cornell Movie Corpus (Danescu-Niculescu-Mizil and Lee, 2011) which contains around 10K movie characters and around 220K dialogues.

**Metrics:** To evaluate the performance and language response quality for each decoder strategy,

<sup>3</sup> $\text{sim}$  does not need to be differentiable.

<sup>4</sup>Brevity penalty introduces diversity and foster longer sentences.

<sup>5</sup>Implementation details are given in supplementary

we use two classical different metrics at the sentence level. (1) A BLEU-4 score (Papineni et al., 2002) is computed on the unigrams, bigrams, trigrams and four-grams; and then micro-averaged. (2) A Diversity score: distinct-n (Li et al., 2016a) is defined as the number of distinct n-grams divided by the total number of generated words. Indeed, in neural response generation, we want to avoid generate generic responses such as ”I don’t know”, ”Yes”, ”No” and foster meaningful responses.

#### 3.2 Response Quality

Figure 1 shows our proposed system results in BLEU-4 score metric. We see that our proposed methods (*BidiS* and *BidiA*) achieve better performances than *VBS* showing that bidirectionality boosts performances.  $BidiA_{BLEU_T}$  achieves the best result overall yielding to a relative improvement of 9% on Cornell and 5% on SWA. From Figure 1, we see that for two different metric  $\text{sim}$  *BidiA* leads to better results than both other algorithm. Improvement of *BidiS* over the baseline *VBS* shows that the optimisation of  $\lambda$  on the validation set leads to good generalisation on the test set.  $BidiA_{BLEU_T}$  is slightly better than  $BidiA_{WMD_T}$  which is likely to be related to the choice of evaluation metric. From Figure 1 we can see that the BLEU-4 score of *VBS* stop increasing when  $B > 10$ . *BidiS* and *BidiA* keep improving the quality of the sequence while more hypothesis are proposed. This suggests that our bidirectional beam search is more efficient at extracting best sentence as the number of hypothesis increases. From Figure 1 we can see that *VBS*, *BidiA* and  $BidiA_{WMD_T}$  present a drop in the performance for a number of hypothesis of 20 and 40: when performing the beam search for 20 hypothesis we observe that the seq2seq is very confident about sentences that lead to lower BLEU-4 score. Those sentences are not considered when  $B < 20$  and better sentences are extracted when the beam size increases.  $BidiA_{BLEU_T}$  does not present this drop of performance this is due to the metric choice (based on overlaps) that selected different sentences from  $BidiA_{WMD_T}$ .

#### 3.3 Rank Analysis

In this section we compare the index returned by *BidiA* and *Best Hypothesis* as shown in Figure 2. Figure 2 illustrates one of the limitation of likelihood based ranking when an off-shell metric is used for evaluation: the very most likely sentences

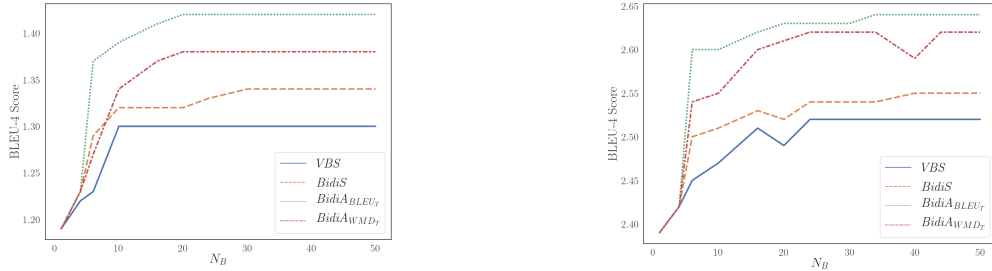


Figure 1: **BLEU-4 Scores** for the proposed algorithms on two different datasets: Cornell (left) and SWA (right).  $N_B$  is the beam size for *VBS* and *BidiS* and 2 times the beam size for *BidiA<sub>WMD<sub>T</sub></sub>* and *BidiA<sub>BLEU<sub>T</sub></sub>*

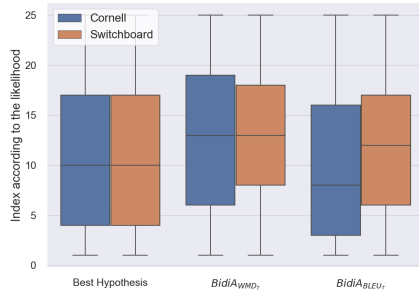


Figure 2: **Index of the response**. Index is the position of the sentence in the beam returned by *VBS*: the most likely sequence is ranked 1, the less likely is ranked 25. *Best Hypothesis* is the sentence (hypothesis) in the beam that yields to the highest BLEU-4.

are not the one with the highest BLEU-4. Interestingly, index distribution of *Best Hypothesis* is very similar for both Cornell and Switchboard, whereas for *BidiA<sub>BLEU</sub>* and *BidiA<sub>WMD<sub>T</sub></sub>* it varies. *BidiA<sub>BLEU</sub>* that has a better BLEU-4 (see Figure 1) than *BidiA<sub>WMD<sub>T</sub></sub>* has an index distribution more similar to *Best Hypothesis* than *BidiA<sub>WMD<sub>T</sub></sub>*.

### 3.4 Diversity of the responses

Table 1 has shown the performance in diversity metrics. Overall, *BidiA* has the best performance among the other strategies (improvement up to 8% over the baseline for Cornell). By looking for an agreement between the reverse seq2seq and the regular one *BidiA* is able to extract sequences that are less likely according to the *VBS*, but more diverse. In all case, we see that bidirectionally helps to have more diverse sentences. Since the influence of the input decreases during the generation bidirectional beam search will output sentences that have both meaningful beginning and ending with respect to the input.

### 3.5 Complexity Analysis

In practical application it is important to evaluate the algorithm complexity when a limited amount of

Model	distinct-n			
	Cornell		Switchboard	
	n=1	n=2	n=1	n=2
<i>VBS</i>	0.051	0.250	0.042	0.231
<i>BidiS</i>	0.051	0.257	0.046	0.240
<i>BidiA<sub>BLEU<sub>T</sub></sub></i>	<b>0.056</b>	0.261	<b>0.050</b>	0.240
<i>BidiA<sub>WMD<sub>T</sub></sub></i>	0.054	<b>0.270</b>	0.048	<b>0.241</b>

Table 1: **Diversity Scores** we report the diversity score (distinct-n) for  $N_B = 50$ .

Algorithm	Complexity
$\mathcal{C}_{VBS}$	$T \times \mathcal{O}(BV \times \log(BV))$
$\mathcal{C}_{BidiS}$	$T \times \mathcal{O}(BV \times \log(BV))$
$\mathcal{C}_{BidiA_{WMD_T}}$	$2T \times \mathcal{O}(\frac{B}{2}V \times \log(\frac{BV}{2}))$
$\mathcal{C}_{BidiA_{BLEU_T}}$	$2T \times \mathcal{O}(\frac{B}{2}V \times \log(\frac{BV}{2}))$

Table 2: **Complexity of the different algorithms**.  $V$  is the size of the dictionary,  $B$  is the beam size,  $T$  is the maximum sentence length.

resources are available. Table 2 shows that *BidiA* is computationally cheaper than *VBS* and that *BidiS* has the same complexity as *VBS*.

## 4 Conclusions

In this paper we show that bidirectional beam search strategies can be leverage to boost the performance of beam search. We have introduced two novel re-ranking criterions that select sentences with more diverse sentences and higher BLEU-4 and reduce computational complexity. Future work includes testing our novel bidirectional strategies with other pretrained models such as the one introduced in (Jalalzai et al., 2020; Chapuis et al., 2021, 2020; Witon et al., 2018), with other types of data (e.g multimodal (Garcia et al., 2019; Colombo et al., 2021a)), on different tasks (e.g style transfer (Colombo et al., 2021b)) as well as exploring other stopping criterions (Colombo et al., 2021c).

## References

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Emile Chapuis, Pierre Colombo, Matthieu Labeau, and Chloe Clavel. 2021. Code-switched inspired losses for generic spoken dialog representations. *arXiv preprint arXiv:2108.12465*.
- Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloe Clavel. 2020. Hierarchical pre-training for sequence labelling in spoken dialog. *arXiv preprint arXiv:2009.11152*.
- Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloe Clavel. 2021a. Improving multimodal fusion via mutual dependency maximisation. *arXiv preprint arXiv:2109.00922*.
- Pierre Colombo, Chloe Clavel, and Pablo Piantanida. 2021b. A novel estimator of mutual information for learning to disentangle textual representations. *arXiv preprint arXiv:2105.02685*.
- Pierre Colombo, Guillaume Staerman, Chloe Clavel, and Pablo Piantanida. 2021c. Automatic text evaluation through the lens of wasserstein barycenters. *arXiv preprint arXiv:2108.12463*.
- Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. 2019. [Affect-driven dialog generation](#).
- Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.
- Tanvi Dinkar, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. 2020. The importance of fillers for text representations of speech transcripts. *arXiv preprint arXiv:2009.11340*.
- Alexandre Garcia, Pierre Colombo, Slim Essid, Florence d’Alché Buc, and Chloé Clavel. 2019. From the token to the review: A hierarchical multimodal approach to opinion mining. *arXiv preprint arXiv:1908.11216*.
- Hamid Jalalzai, Pierre Colombo, Chloé Clavel, Eric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. 2020. Heavy-tailed representations, text polarity classification & data augmentation. *arXiv preprint arXiv:2003.11593*.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. [A persona-based neural conversation model](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003. Association for Computational Linguistics.
- Zachary Chase Lipton. 2015. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective approaches to attention-based neural machine translation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. [The natural language cathlon: Multitask learning as question answering](#).
- Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2018. Forward-backward attention decoder. In *INTERSPEECH*.
- Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290. ACL.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

- Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, Carol Van Ess-Dykema, et al. 1998. Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, pages 3104–3112, Cambridge, MA, USA. MIT Press.
- Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence - video to text. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics.
- Wojciech Witon, Pierre Colombo, Ashutosh Modi, and Mubbasir Kapadia. 2018. Disney at iest 2018: Predicting emotions using an ensemble. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 248–253.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.
- Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1108–1117.

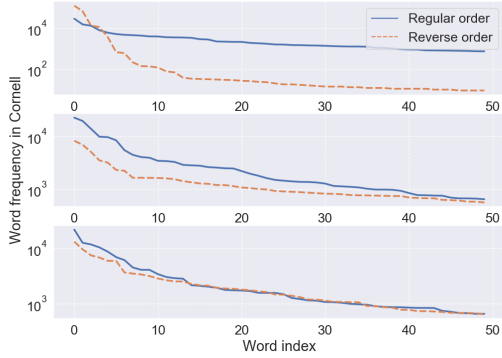


Figure 3: **Word distribution on Cornell for different word position in the sentence.** This figure shows the frequency of appearance of the 50 most common word in the sentence extracted from Cornell. An example of the regular order sentence is "The cat is red.", the associated reverse order is: ". red is cat The". Top plot shows the frequency of words appearing in position 1, the second plot for words appearing in position 2, the third plot for words appearing in position 3.

## 5 Supplementary Material

### 5.1 Corpus analysis: importance of reverse model

In this section, we discuss the importance of using a reverse model. Figure 3 shows the word distribution on Cornell for 50 most common words at each position. From top plot we observe that the regular seq2seq faces a lot of likely choices: all the fifty most common words appear more than  $10^3$  times in position 1. The reverse seq2seq faces less than 5 very likely choices in position 1 that appear more than  $10^3$  time. The reverse seq2seq is then less likely to transmit a mistake at time step 2.

### 5.2 Ideal reranking

In Table 3 we report the BLEU-4 achieved by VBS and *Best Hypothesis*: the best hypothesis alive in the beam. It illustrates that the limitations of the likelihood criterion and shows that making a change in the final reranking and sentence selection criterion can yield to higher BLEU-4.

### 5.3 Implementation of similarity measure ( $sim$ )

In this section we describes the implementation of each similarity  $sim$  used in section 3. We have introduce a brevity penalty for two main reasons:

- preliminary experiments have shown that the

		BLEU-4				
		Beam Size	1	6	10	50
Corn.	VBS		1.19	1.23	1.30	1.30
	<i>Best Hypoth.</i>		1.19	1.40	1.60	2.56
SWA	VBS		2.39	2.45	2.47	2.52
	<i>Best Hypoth.</i>		2.39	3.40	4.30	5.77

Table 3: **BLEU-4 Scores on Cornell (Corn.) and Switchboard (SWA):** VBS stands for the standard beam search (see section 2) *Best Hypoth.* is the hypothesis in the beam that leads to the highest BLEU-4. In our work performances of *Best Hypoth.* can be seen as an upper bound of the performances of VBS.

regular seq2seq tends to generate short sentences due to the data distribution.

- if no brevity penalty is introduced and both neural networks generate "I don't know" the selected sentence will be "I don't know" since the similarity measure will be 1. With a brevity penalty, similarity metric can select a less generic choice.

#### 5.3.1 $BLEU_T$

$BLEU_T$  has been implemented by using the nltk library <https://www.nltk.org/>. In Equation 9 we set  $sim = 1 - BLEU_T$ .

#### 5.3.2 $WM_T$

$WM_T$  uses the wm-relax library (<https://github.com/src-/wmd-relax>), embeddings used are coming from FastText library (Bojanowski et al., 2017). At the first step stopwords according nltk list are removed, Word Mover Distance is computed and multiplied by  $BP_T$  previously defined. Formally, in Equation 9 we set  $sim = WM_T$ .

### 5.4 Architecture details

We evaluate our proposed algorithms by using off-the-shelf seq2seq models. For the encoder, we use two-layer bidirectional GRU (Chung et al., 2014) (256 hidden layers). For the decoder, we use a one-layer uni-directional GRU (512 hidden layers) with attention (Luong et al., 2015). The embedding layer is initialized with fastText pre-trained word vectors (on Wikipedia 2017, the UMBC web-based corpus and the statmt.org news dataset) and the size is 300 (Bojanowski et al., 2017). We use the ADAM optimizer (Kingma and Ba, 2014) with a

learning rate of 0.001, which is updated by using a scheduler with a patience of 100 epochs and a decrease rate of 0.5. The gradient norm is clipped to 5.0, weight decay is set to  $1e^{-5}$ , and dropout (LeCun et al., 2015) is set to 0.1. The models have been implemented with pytorch, they have been trained on 97%, validated on 1%, and tested on 2% of the data respectively. Since our purpose is to show that bidirectionality can boost beam search we set  $\alpha = \frac{1}{2}$  in Equation 2.1.

## 5.5 Proofs of Complexity analysis

### 5.5.1 VBS complexity

For VBS, at each time step  $BV$ , hypotheses are re-ranked and the  $B$  most likely are kept. The final average complexity is:

$$\mathcal{C}_{VBS} = T \times \mathcal{O}(BV \times \log(BV)) \quad (6)$$

### 5.5.2 BidiS complexity

In the case of *BidiS*, the algorithm generates  $B$  sequences using VBS, and then for generating sequence  $Y_T$  it computes  $\frac{\log P(Y_T^-|X)}{lp(Y_T)}$  with complexity  $\mathcal{O}(T)$ . The final step includes a sorting of complexity  $\mathcal{O}(B \log(B))$ .<sup>6</sup> *BidiS* complexity is:

$$\mathcal{C}_{BidiS} = T \times \mathcal{O}(BV \times \log(BV)) \quad (7)$$

### 5.5.3 BidiA complexity

**Word Mover’s Distance criterion:** According to (Kusner et al., 2015) the computational cost of the Word Mover’s Distance computation is  $\mathcal{O}(p^3 \times \log(p))$ , where  $p$  denotes the number of unique words in the documents. In our case the distance is computed between two sequences of length at most  $T$ , hence  $p \leq 2T$ . *BidiA*<sub>WMD<sub>T</sub> complexity with Word Mover’s Distance as selection criterion is given by the following formula:</sub>

$$\begin{aligned} \mathcal{C}_{BidiA_{WMD_T}} &= 2T \times \underbrace{\mathcal{O}\left(\frac{B}{2}V \times \log\left(\frac{BV}{2}\right)\right)}_{\text{two VBS with with beam size } \frac{B}{2}} \\ &+ \frac{B^2}{8} \times \underbrace{\mathcal{O}(8T^3 \times \log(2T))}_{\text{pairwise WMD}} \quad (8) \\ &+ \underbrace{\mathcal{O}(T)}_{\text{complexity of } BP_T} \end{aligned}$$

In general  $T^3 \leq BV$  and  $T \lll BV$ , in Equation 9 the second term is small compared to the first

<sup>6</sup> $\mathcal{O}(B \log(B))$  and  $\mathcal{O}(T)$  have much less order compared to  $\mathcal{O}(BV \times \log(BV))$  so they can be neglected here.

term, hence  $\mathcal{C}_{BidiA_{WMD_T}} \approx T \times \mathcal{O}(BV \times \log(\frac{BV}{2}))$ . Even though  $V$  dominates the complexity of the algorithm, still *BidiA*<sub>WMD<sub>T</sub> is more efficient than VBS.<sup>7</sup></sub>

**BLEU criterion:** the computational cost of the *BLEU*<sub>T</sub> score is polynomial in  $T$ . *BidiA*<sub>BLEU<sub>T</sub></sub> complexity with BLEU score as the selection criterion is given by the following formula:

$$\mathcal{C}_{BidiA_{BLEU_T}} = 2T \times \mathcal{O}\left(\frac{B}{2}V \times \log\left(\frac{BV}{2}\right)\right) \quad (9)$$

<sup>7</sup>For example if  $T = 30$ ,  $B = 30$ ,  $V = 35k$  we see that  $\mathcal{C}_{VBS} = 1.4 \times \mathcal{C}_{BidiA}$ .