Learning to Organize a Bag of Words into Sentences with Neural Networks: An Empirical Study

Chongyang Tao[1], **Shen Gao**[1], **Juntao Li**[1], **Yansong Feng**[1], **Dongyan Zhao**[1*], **Rui Yan**[1,2,3*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Beijing Academy of Artificial Intelligence
[1]{chongyangtao,shengao,lijuntao,fengyansong,zhaody}@pku.edu.cn
[2]{ruiyan}@ruc.edu.cn

Abstract

Sequential information, a.k.a., orders, is assumed to be essential for processing a sequence with recurrent neural network or convolutional neural network based encoders. However, is it possible to encode natural languages without orders? Given a bag of words from a disordered sentence, humans may still be able to understand what those words mean by reordering or reconstructing them. Inspired by such an intuition, in this paper, we perform a study to investigate how "order" information takes effects in natural language learning. By running comprehensive comparisons, we quantitatively compare the ability of several representative neural models to organize sentences from a bag of words under three typical scenarios, and summarize some empirical findings and challenges, which can shed light on future research on this line of work.

1 Introduction

Though significant progress has been made, it is still mysterious how humans are able to understand, organize, and generate natural languages. In the field of natural language processing, many efforts have been made to enhance computational models. Recently, recurrent neural networks (Mikolov et al., 2010) and encoder-decoder architectures (Sutskever et al., 2014) with long short-term memory (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (Chung et al., 2014) have demonstrated state-of-the-art performance in sequence modeling and generation.

Nowadays, the encoder-decoder architectures have become a widely used approach for sequence-to-sequence tasks such as machine translation (Bahdanau et al., 2015), text summarization (Paulus et al., 2018) and dialogue generation (Serban et al., 2016). Such models generally encode the input sequence into a vector representation using recurrent neural networks (RNNs) (Sutskever et al., 2014), convolutional neural networks (Gehring et al., 2017) or transformer architectures (Vaswani et al., 2017). The decoder then produces the output sequence step-by-step, conditioned on the encodings of the encoder. Basically, those encoders process information along the sentence sequences, where sequential information is recurrently modeled at each position of the sequences. Thus these models are sensitive to word orders. Moreover, it has been demonstrated that order matters in sequence encoding (Vinyals et al., 2015). Admittedly yes, order information is important for sequences learning and encoding. An interesting question might be that, is it possible to encode natural languages without considering order information?

Take a look at an example of word rearrange quizzes for language learners[1]. Given a bag of words from a disordered sentence {*the dog James talking sat next to himself to .*}, most people can still read with little effort, though disagreement might exist on subtle details as to whether it is the man or the dog that is seated. Inspired by this, it is interesting to explore how and to what extent we can encode natural languages without considering order information.

From a computational perspective, we ask: *Can we construct an algorithm that is capable of reading a bag of words as robustly as humans do?* Our task is to predict the original sentence given a bag of words without orders extracted from a random sentence. This orderless setting is important to characterize the human instinct for understanding languages. The answer to this question also provides insights into many important practical problems: In abstractive text summarization, the summary can be generated according to a bag of extracted key words (Xu et al., 2010); In statistical machine translation, we need to reorder the words or phrases in the target language to get a natural and fluent

[*] Corresponding authors: Dongyan Zhao and Rui Yan.

[1] https://quizlet.com/143171956/arrange-words-and-form-meaningful-sentences-flash-cards/

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1682–1691
June 6–11, 2021. ©2021 Association for Computational Linguistics

Normal	**Input**: the dog James talking sat next to himself to . **Output**: James sat next to the dog talking to himself .
Noise	**Input**: the rule dog James talking sat next to himself to . dashed **Output**: James sat next to the dog talking to himself .
Missing	**Input**: dog James talking next to himself to . **Output**: James sat next to the dog talking to himself .

Table 1: 3 scenarios for sentence organization. Words marked in red denote the added noisy words, and words marked in green denote the missing words.

sentence (He and Liang, 2011). In dialogue systems, we need systems that are enabled to converse smoothly with people that have troubles in ordering words, such as children, language learners, and speech impaired. In image caption, the caption can be organized with a bag of attribute words extracted from the image (Fang et al., 2015). Moreover, such a model can help non-native speakers of English to write a sentence just from keywords.

This bag-to-sentence transformation problem is rather challenging primarily due to three reasons. First, the relationship between words is missing from the input bag of words. To predict the correct ordering, both the meaning of the whole sentence and the words that may become the context of a particular word must be guessed and leveraged. Second, the input bag of words might only be a subset of all the words in a sentence, and there might exist randomly injected words, as shown in Table 1. Last, the correct ordering of the words into a sentence may not be unique, and the model needs to have the flexibility to allow multiple choices of outputs.

While much research has been directed into processing sequential text information, there has been far less research regarding the encoding of an unordered bag. A simple approach is based on pooling that takes the maximum value for each dimension of the word embeddings (Qi et al., 2017). This strategy is effective in simple tasks (e.g., sentence classification) but loses much contextual information for sentence organization. (Vinyals et al., 2015) proposes to encode a set through iterative attention on the input items, alike to the memory network. These approaches could obtain an order-invariant representation of the set from a global perspective. However, they are lacking of modeling the semantic dependencies between input items. In addition, the effectiveness of these models on the bag-to-sentence transformation problem is also unknown.

In this paper, we aim to investigate how "order" information takes effects in natural language learning for neural models. On the basis of the pooling-based and memory-based approaches, we introduce the self-attention to encode the semantic dependencies between input words without considering order information, so as to enrich individual words with contextual information from different semantic aspects. We systematically compare the ability of different neural models to organize sentences from a bag of words in terms of three typical scenarios shown in Table 1. The contributions of this paper are summarized as follows:

- We present an empirical study to investigate the ability of neural models to organize sentences from a bag of words.
- We introduce a bag-to-sentence transformation model based on self-attention, which significantly outperforms existing models in sentence organization tasks.
- We show some interesting results by thoroughly comparing and analyzing sentence organization under different scenarios (*Normal, Noise, Missing*), which may shed light on future research on this line of work.

2 Related Work

Pooling is a basic approach to encode sets (or bags), and has been widely used for many tasks, such as 3D shape recognition (Qi et al., 2017), few-shot image classification (Snell et al., 2017). Besides, several studies have explored the capability of attention mechanisms in modeling sets (or bags). Vinyals et al. (2015) proposed to encode a set with multi-hop attention operations. Ilse et al. (2018) proposed to use attention-based weighted sum-pooling for multiple instance learning. Similarly, Yang et al. (2020) proposed an attention-based algorithm to aggregate a deep feature set for multi-view 3D reconstruction.

As a new approach to modeling a text sequence, self-attention has been successfully used in many NLP tasks, such as machine translation (Vaswani et al., 2017), text summarization (Paulus et al., 2018) and machine reading comprehension (Wang et al., 2017). However, most studies about self-attention focus on sequence modeling, which ignores the positional invariance of the attention mechanism itself. In perticular, Ma et al. (2018) utilized self-attention to model interactions between the objects in a video, and employed pooling to obtain aggregated features. On this basis of the transformer architecture, Lee et al. (2019) presented an Set Transformer designed to model interactions among elements in the input set.

Without considering missing words or noisy words, our task devolves into word ordering problem, which is a fundamental task in natural language generation. Previous, researchers usually employed N-gram based language models (De Gispert et al., 2014; Schmaltz et al., 2016), syntactic-based language models (Zhang and Clark, 2011; Liu et al., 2015) or combined models (Zhang et al., 2012; Liu and Zhang, 2015) to solve this problem. More recently, Hasler et al. (2017) proposed a bag-to-sequence model, where the decoder RNN directly attended to the word embeddings. However, all these methods aim at finding the best permutation of a bag of words based on language models, and do not consider how to encode a bag of words.

3 Problem Formulation

Given a bag of words $X = \{x_1, x_2, \cdots, x_m\}$ which consists of m tokens, our model will generate a sentence $Y = \{y_1, y_2, \cdots, y_n\}$, where n is the length of target sentence. In the normal scenario, the words of X come from a disordered sentence and are the same as Y. While in other two scenarios, the condition no longer holds. To be specific, X contains some noisy words that do not appear in Y for noise scenario, and X lacks some words that should appear in generated sequence for the missing scenario. We can model this using the conditional probability $P(Y|X)$ and decompose it with the chain rule.

$$P(Y|X) = \prod_{t=1}^{n} P(y_t|y_1, y_2, \cdots, y_{t-1}, X), \quad (1)$$

In our scenario, the source input is a bag of words or even with noisy or missing words and the output is a sentence.

4 Bag-to-Sequence Models

In this paper, we employ encoder-decoder frameworks to address the bag-to-sentence problem. Particularly, the encoder is responsible for learning an order-invariant context representation for the input bag, and the decoder produces the target sentence conditioned on a bag of input words.

4.1 Compared Encoders

We consider four representative neural models to encode the unordered bag of words as follows.

RNN. Recurrent neural networks typically process information along the word positions of the input sequence, and they have proven to be sensitive to variations of word order to some degree (Vinyals et al., 2015). In this paper, we introduce an RNN with long short-term memory units (LSTMs) as a baseline encoder for a comparison. Formally, the hidden state of RNN at the t-th step $\mathbf{h}_t$ is calculated by:

$$\mathbf{h}_t = \mathtt{LSTM}(\mathbf{h}_{t-1}, \mathbf{w}_t), \quad (2)$$

where $\mathbf{w}_t$ denotes the input word embedding at t-th step. The final hidden state of LSTM is regarded as the context representation of the input bag.

Pooling. A simple way to encode a bag without considering order information is the pooling-based approach as inspired by Qi et al. (2017) that summarizes bag information by choosing the maximum value from each dimension of the word embeddings. Formally, given a bag of word embeddings $\{\mathbf{w}_i\}_{i=1}^{n}$, the context representation of the input bag of words $\mathbf{v}_s$ can be calculated as:

$$\mathbf{v}_s = \mathtt{max}\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}, \quad (3)$$

Memory. The memory-based approach encodes a bag of words through performing multiple rounds of attention over the word representations, alike to the memory network (Sukhbaatar et al., 2015). Formally, we take the vector representation $\mathbf{v}_s$ obtained by the pooling-based method as the initial bag representation $\mathbf{v}_s^0$. At the t-th processing round, we use the current bag representation $\mathbf{v}_s^t$ to attend the memory $\{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$ composed of word embeddings, and compute an attention vector $\mathbf{r}_t$ through the attention mechanism (Bahdanau et al., 2015), defined as:

$$\begin{aligned} \alpha_{t,i} &= \frac{\exp(g(\mathbf{v}_s^t, \mathbf{h}_i))}{\sum_{i=1}^{n} \exp(g(\mathbf{v}_s^t, \mathbf{w}_i))}, \\ \mathbf{r}_t &= \sum_{i=1}^{n} \alpha_{t,i} \mathbf{w}_i, \end{aligned} \quad (4)$$

where $g(\cdot,\cdot)$ is a function that computes the similarity between $\mathbf{w}_i$ and $\mathbf{v}_s^t$, and we employ dot product function in this paper. Then the current bag representation $\mathbf{v}_s^t$ is concatenated with the output of the attention vector $\mathbf{r}_t$, and further transforms it through non-linear transformation.

$$\mathbf{v}_s^{t+1} = f([\mathbf{v}_s^t; \mathbf{r}_t]), \quad (5)$$

where $f(\cdot)$ is a non-linear mapping function which reduces the input dimension to d_e. Following Vinyals et al. (2015), we use an LSTM unit (Hochreiter and Schmidhuber, 1997) (without inputs) as $f(\cdot)$. We perform this process for K rounds. The obtained vector $\mathbf{v}_s^K$ is the final bag representation. We set K as the number of tokens in source bag.

Self-attention. Self-attention is a special case of standard attention mechanism (Bahdanau et al., 2015) where each word can attend to (interact with) all words in the input. Unlike RNNs, self-attention can model dependencies among words in the input bag without considering the order information. In this paper, we borrow the idea from the work of neural transformer architecture (Vaswani et al., 2017). The model contains N stacked blocks, each of which mainly composed of a multi-head attention layer and a row-wise feed-forward layer. More compactly,

$$\{\mathbf{m}_1, \cdots, \mathbf{m}_n\} = \texttt{MultiHeadAtt}(\{\mathbf{w}_1, \cdots, \mathbf{w}_n\}), \tag{6}$$

$$\{\mathbf{h}_1, \cdots, \mathbf{h}_n\} = \texttt{FFN}(\{\mathbf{m}_1, \cdots, \mathbf{m}_n\}), \tag{7}$$

where $\mathbf{m}_i$ and $\mathbf{h}_i$ are the representation for i-th word produced by the multi-head attention layer and the row-wise feed-forward layer respectively. A residual connection (He et al., 2016) and a row-wise normalization (Ba et al., 2016) are applied around each of the multi-head attention layer and feed-forward layer.

Based on the representation produced by the self-attention, we further employ pooling-based or memory-based approaches[2] to obtain a global context representation for input bag. We name the full model as **AttP** when *pooling-based* approach is adopted, and name it as **AttM** by using *memory-based* approach.

4.2 Decoder

The decoder acts as a language model to reconstruct the sentence conditioned on the bag representation. To highlight the differences among different encoders, we utilize the same decoder for different encoders.

Since the target Y corresponds to a sequence, and has significant vocabulary overlap with the input bags of words, we blend a pointer-based decoder (Vinyals et al., 2015; See et al., 2017), which acts as a language model to enable our model to generate a word from the vocabulary, or to copy words from the input via the pointer mechanism. Particularly, to calculate the context vector $\boldsymbol{c}_t$ and pointer probabilities in each decoding step, we take the input word embeddings as the hidden states in pooling- and memory-based approaches. In self-attention-based approaches, we take the output representations of the self-attention layer as the hidden states.

4.3 Objective Function

Our goal is to maximize the output sentence probability given the input bag of words. Therefore, we optimize the negative log-likelihood loss function:

$$J(\Theta) = -\frac{1}{\mathcal{D}} \sum\nolimits_{(x,y)\in\mathcal{D}} \log p(y|x), \tag{8}$$

where $\mathcal{D}$ is a set of bag-sentence pairs and Θ is the parameters.

5 Experiments

We evaluate our method quantitatively and qualitatively on a large dataset for three typical sentence organization scenarios described in Table 1.

5.1 Datasets

We construct a large dataset from The Wesbury Lab Wikipedia Corpus[3] (Shaoul, 2010), which is created from the articles in English Wikipedia. We tokenize all articles into sentences using the NLTK package[4], and replace all numbers with “__num__". We retain experiment samples among the sentences of length between 5 and 20 to focus on the majority case of the training corpus. Finally, we randomly sample 10 million sentences for training, 100k for validation and 10k for testing. In the normal scenario, we randomly shuffle the words in each sentence as the input of our model, and the original sentence is the ground truth. Based on the normal scenario, we construct the training data for the noise scenario by randomly introducing some noisy words to the source bag, and construct the training data for the missing scenario by randomly removing some words from the source bag.

We also compare the normal scenario of our model on The English Penn Treebank

[2] It is worth noting that the current memory is composed of the word representations output by self-attention layer

[3] The corpus removes all links and other irrelevant material (e.g., navigation text, etc), and contains about one billion words, over 2 million documents.

[4] http://www.nltk.org/api/nltk.tokenize.html

	BLEU			ROUGE-L			*Perfect Matching Rate (PMR)*			*Word Accuracy (WAcc)*		
	Normal	Noise	Missing	Normal	Noise	Missing	Normal	Noise	Missing	Normal	Noise	Missing
Pooling	0.4656	0.4382	0.2636	0.6917	0.6587	0.5470	0.1945	0.1461	0.0426	0.5685	0.5536	0.4437
LSTM	0.4736	0.4327	0.2538	0.7311	0.6761	0.5453	0.2203	0.1542	0.0390	0.5808	0.5563	0.4369
Memory	0.5030	0.4537	0.2664	0.7485	0.6939	0.5607	0.2404	0.1672	0.0450	0.6063	0.5789	0.4520
AttP	0.5740	0.5372	0.2882	0.7860	0.7396	0.5722	0.3014	0.2267	0.0479	0.6613	0.6367	0.4700
AttM	**0.5886**	**0.5433**	**0.2914**	**0.7925**	**0.7465**	**0.5738**	**0.3208**	**0.2355**	**0.0512**	**0.6697**	**0.6461**	**0.4702**

Table 2: Results on the test sets of three scenarios for Wikipedia dataset. We randomly generate noisy words with the number between 1 and half length of the sentence from the vocabulary for each sentence as the input of the the noise scenario. For the missing scenario, random words with number between 1 and half length of the sentence are removed from each sentence. It is worth noting that we randomly shuttle input bags with three different seeds and report the mean score of each metrics for LSTM.

data (PTB) (Marcus et al., 1993), which is a widely-used dataset for word ordering task (Schmaltz et al., 2016; Hasler et al., 2017). To facilitate fair comparisons, we use the data preprocessed by (Schmaltz et al., 2016), which consists of 39, 832 training sentences, 1, 700 validation sentences and 2, 416 test sentences.

5.2 Implementation Details

For all models, we set the dimension of word embedding as 128. In the LSTM-based encoder, the dimension of hidden unit is 256. In the self-attention-based encoder, we set the number of head in Equation (6) as 8 and the hidden size of feed-forward layer in Equation (7) as 256. All parameters are tuned in the validation set. The vocabulary size is 50k. We use AdaGrad (Duchi et al., 2011) optimizer on mini-batch of size 32, with learning rate at 0.15 and gradient clipping at 2. In decoding, we set the beam size as 5 for all models. It is worth noting that we do not compare with the results derived from the modified beam search method proposed in Hasler et al. (2017) since we focus on investigating the capability of a model to encode a bag of words in this paper. So we compare all methods under standard beam search method (with a beam size of 5) in our experiment, to highlight the differences among different encoders.

5.3 Evaluation Metrics

In our settings, a shuffled sentence sometimes may correspond to multiple reasonable outputs. Hence we employ four automatic evaluation metrics to evaluate the quality of a generated sentence from different aspects. **PMR** (*Perfect Matching Ratio*) measures the ratio of instances that are exactly the same as the ground-truth. **BLEU** (Papineni et al., 2002) measures the quality of generated sentences by computing overlapping lexical units (e.g., unigram, bigram) with the reference sentences. **ROUGE-L** (Lin, 2004) measures the longest common subsequence (LCS) between the reference sentence and the generated sentence. **WAcc** (*Word Accuracy*) is the negative word error rate (WER) (Mangu et al., 2000). It measures the edit distance between the generated sentence and the reference sentence (higher is better). Besides, we also conduct human evaluations to further analyze our generated results and explore the detail sort of wrong cases.

	BLEU	ROUGE-L	WAcc	PMR
N-GRAM*	0.2330	-	-	-
RNNLM*	0.2450	-	-	-
Pooling	0.3118	0.5916	0.4105	0.0863
LSTM	0.3140	0.5875	0.3873	0.0850
Memory	0.3328	0.6053	0.4089	0.0941
AttP	0.3469	0.6169	0.4297	0.1013
AttM	**0.3489**	**0.6194**	**0.4304**	**0.1059**

Table 3: Results of word ordering task on PTB datasets (beam size = 5), * denotes the results reported in (Hasler et al., 2017).

5.4 Overall Results

Table 2 illustrates the performance of all models for three scenarios on the Wikipedia dataset. Firstly, we can find that *Pooling* shows the worse performance among all models. This is because directly utilizing pooling operation on word embeddings would lose track of much crucial context information. Secondly, although *LSTM* processes the information sequentially, it achieves better results than *Pooling* in normal and noise scenarios. A possible explanation for this might be that the parameters in *LSTM* enable the mode to retain some bag information.

In particular, self-attention-based approaches (e.g., *AttP* and *AttP*) show the best results, and outperform *Memory* by a large margin in terms of

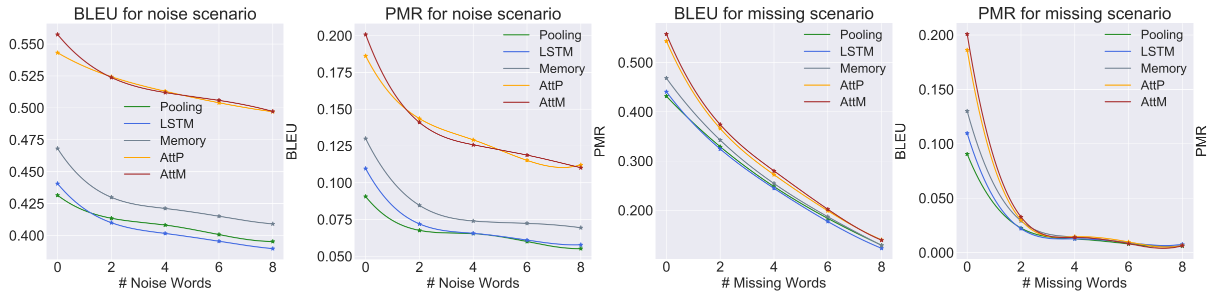

Figure 1: Performance in terms of different metrics by varying the number of missing words or noisy words. We continuously introduce noisy words or missing words with the footstep of 2. The noisy words are randomly picked from the vocabulary.

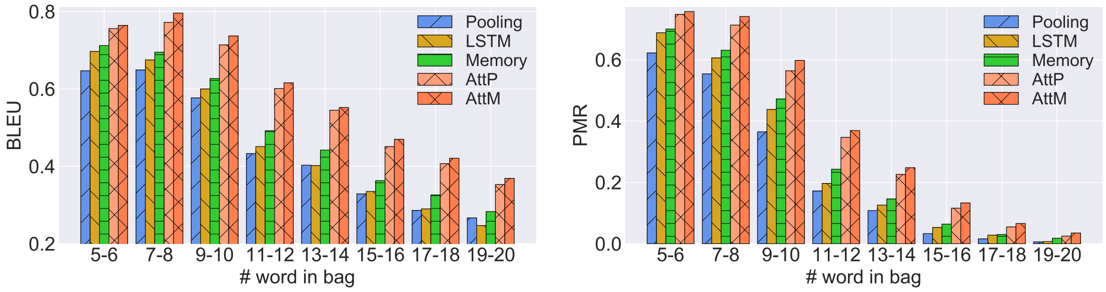

Figure 2: Performance in terms of different metrics by varying the number of words in source bag.

all evaluation metrics, especially for normal and noise scenarios. The phenomenon might be ascribed to the reason that *Memory* encodes the bag of words by considering each word individually, while *Self-attention* captures the semantic dependencies among the input words from different semantic aspects, leading to a more robust bag representation. Additionally, *AttM* shows better performance than *AttP*, indicating that the memory-based fusion method is more useful than the pooling-based fusion method.

In addition, we can notice that the performance of all models declines when noisy words are introduced or some words are removed from the input bag, but much more for removing some words. This result may be explained by the fact that organizing a sentence from a partially observable bag of words is more challenging since it requires background knowledge to predict the meaning of the bag and further fill the missing words. On the other hand, in the noise scenario, most noise words have a small impact on learning the context representation of a bag and all words can be decoded (or generated) via copy operations.

We further run experiments on the PTB dataset, which is a benchmark for the word ordering task. The results are shown in Table 3. We can observe that various neural models outperform the traditional N-GRAM model and RNNLM. In these neural models, the results are consistent with those of Wikipedia.

5.5 Discussions

The impact of the number of noisy/missing words. To better understand the robustness of different models under the noise scenario and the missing scenario, we show how the performance changes as the number of noise or missing words changes in Figure 1. As seen, approaches based on self-attention always outperform other approaches in both scenarios, especially more significantly in the noise scenario. Besides, the performance of all models drops as the increases of the number of missing words or noisy words, but more sharply for the missing scenario. The results imply that: 1) In the bag-to-sentence transformation problems, the capability of neural models to resist noisy words is better than the capability to resist missing words; 2) It is still challenging for neural models to handle the bags where some information is missed.

The impact of bag size. We further study how the size of the input bag influences the performance of different models. Figure 2 illustrates how the

			$\log p(y\|x)$
Case-1	**Input**:	. largest animals bears the they in the land also are only native taiwan and	
	Reference:	they are also the largest land animals and the only native bears in taiwan .	
	Beam-1:	they are also the only native animals and the largest land bears in taiwan .	-0.2602
	Beam-2:	they are also the only native land animals and the largest bears in taiwan .	-0.2708
	Beam-3:	they are also the largest land animals and the only native bears in taiwan .	-0.3183
Case-2	**Input**:	a , engineering there . time mechanical chairman long he for served , as of	
	Reference:	there he served , for a long time , as chairman of mechanical engineering .	
	Beam-1:	there , he served as chairman of mechanical engineering , for a long time .	-0.0797
	Beam-2:	there , he served for a long time , as chairman of mechanical engineering .	-0.0882
	Beam-3:	for a long time , there , he served as chairman of mechanical engineering .	-0.2041
Case-3	**Input**:	their cuddy again however . sends interrupts and , exchange away ali	
	Reference:	however , cuddy interrupts their exchange again and sends ali away .	
	Beam-1:	however , ali interrupts their exchange again and sends cuddy away .	-0.1829
	Beam-2:	however , cuddy interrupts their exchange again and sends ali away .	-0.2116
	Beam-3:	however , cuddy interrupts their exchange and sends ali away again .	-0.2187

Table 4: Generation examples of 3 different results via beam search in our *AttM* under normal scenario. We show the log generation probability for each beam candidate in the last column.

performance of *AttM* changes with respect to bags with different numbers of words in the normal scenario, where we bin test examples into buckets. We observe a similar trend for all models: they first remain stable when the bag size less than 8, and then decrease monotonically when the bag size keeps increasing. The reason might be that when only a few words are available in input bag, the model can well capture the meaning of the whole sentence, but when the bag becomes large enough, the semantic combination of words will become more complicated and the meaning of target sentence will be hard to be grasped. Besides, self-attention-based models always achieve the best performance, which is consistent with the result in Table 2.

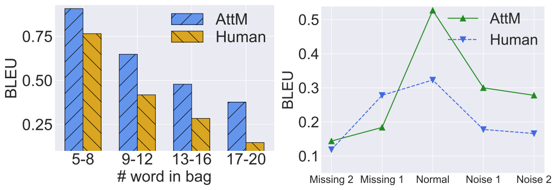

Figure 3: Performance of the neural model and human in terms of different scenarios.

Multiple plausible outputs. Actually, for the bag-to-sequence task when applied to language, a bag of words sometimes may correspond to multiple reasonable and grammatical outputs. Such a phenomenon is similar to response generation in dialog systems, where several responses can be reasonable. Table 4 shows the three generated results of *AttM* (the most strong model) through beam search. We can notice that all generated sentences are grammatical and reasonable. In case-1, the objects “animals” and “land bears” are exchangeable in terms of syntax; both “native” and “largest” can describe these objects. Our model prefers “the only native animals” and “the largest land bears”. Since our model is a conditional language model learned from the training corpus, and the decoder reconstructs a sentence conditioned on the representation of the input bag of words. The joint probability of sentence-1 is larger than sentence-2. In case-2, “for a long time” and “there” are adverbials, and are position variable. However, the meaning of all generated sentences remains the same. In case-3, both “ali” and “cuddy” are names, thus they are undistinguishable in this situation. Our model assigns a higher probability to “ali interrupts their exchange again and sends cuddy away”. Despite the lack of order information, neural models can still organize all possible sentences through beam search.

Neural Models Vs. Human. We are also curious about the ability of humans to organize sentences from a bag of words. We first binned the test set of the normal scenario into four buckets according to the size of the input bag, and then randomly selected 40 samples from each bucket. We invited humans to organize the target sentence regarding the input bag using crowd-sourcing. Each bag was randomly presented to 3 judges and we retain the answer with the highest BLEU score. Figure 3(a) illustrates the BLEU score of humans and the most competitive model AttM across different bag sizes. We observe that both the performance of humans and AttM become worse with the increase of the bag size, which is consistent with the result in Figure 2. Besides, AttM always shows better performance than human, but the performance gap

	Annotated Types	*Ratio*
Synonymous (30%)	Exactly generated	16%
	Two adverbials are exchanged	5%
	Two coordinate clauses are exchanged	7%
	Other reasons	2%
Non-synonymous (57%)	The subject and object are exchanged	5%
	The logic is unreasonable.	33%
	Other reasons	19%

Table 5: The statistical analysis of 100 randomly selected samples for *AttM* in normal scenario. We only show the result of the grammatical part, and the proportion of ungrammatical samples is 13%.

becomes smaller as the bag size decreases. This result indicates that humans are better at recognizing small bags than large bags.

Besides, we also study how noisy words and missing words impact the performance of humans and neural models. Based on the above test set randomly selected from the normal scenario, we randomly introduced 1 or 2 noisy words to the source bag denoting as *noise-1, noise-2* respectively, and randomly removed 1 or 2 words from the source bag, denoting as *missing-1, missing-2* respectively. We also invited humans to organize a sentence regarding the input bag using crowdsourcing. Figure 3(b) presents the results of each test set. We summarize our observations as follows: (1) Both the performance of human and AttM get worse when noisy words are introduced or some words are removed; (2) Compared with neural models, humans are more robust to noisy words and missing word in sentence organization; (3) The performance AttM is significantly better than humans, but becomes comparable with humans when 2 words are randomly removed from the input bag. The results imply that humans have a more strong background knowledge of language to guess the meaning of the target sentence and complete the cloze test.

Error analysis. To further analyze the quality of the generated sentence and in which case our model fails to recover the original sentence, we invite four educated annotators to judge the quality of 100 randomly sampled sentences[5] generated by *AttM*. Annotators were asked to judge whether a generated sentence is grammatical and the meaning of a generated sentence is the same as the ground truth. We can find that 87% of generated sentences are grammatical and 30% of sentences share the same meaning with the ground-truth. Among those grammatical and synonymous samples, 46.7% (14/30) of sentences are not exactly the same with the ground truth in syntax. There are two main types of paraphrase: the position of adverbials is exchanged or the position of coordinate clauses is exchanged. Among those grammatical and non-synonymous samples, the logic of the majority sentences is unreasonable due to the position exchange of adverbials or coordinate clauses, and unreasonable combinations of semantic units. Besides, the semantics of some sentences are changed because of the exchange of the subject and the object.

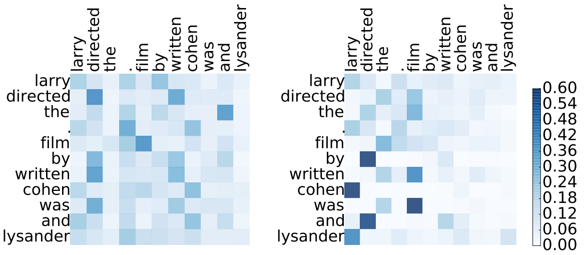

Figure 4: Visualization of attention weight in the 5-th head (left) and 6-th head (right) in self-attention. The target sentence is "the film was written and directed by larry cohen ." "lysander" is a noisy word.

Attention visualization. Figure 4 shows the visualization of attention weights of different heads in the 5-th block from the self-attention layer. We can observe that self-attention can capture combinatorial relations between the input words from different semantic aspects. For instance, "cohen" shows a strong correlation with "larry" in both heatmaps since "Larry Cohen" is the name of a famous director. Moreover, both "was" and "by" attend to "directed" and "written", composing the phrase "was written (directed) by". Such combinatorial relations can make the word representation more informative, which contributes to the representation learning of the bag of words. Additionally, we observe that almost all words but itself demonstrate weak correlations with the noisy word "lysander" in both heatmaps, demonstrating the advantages of our model to tolerate noisy words.

6 Conclusions

In this paper, we present an empirical study to investigate the ability of neural models to organize sentences from a bag of words under three typical scenarios. We conclude our discussion with the following findings:

- Self-attention is effective to capture the se-

[5] We randomly select samples with a bag size greater than or equal to 10 since they contain more error cases.

mantic dependencies between words in the input bag and shows competitive performance in bag-to-sentence transformation.

- Neural models have a certain degree of capability to organize a sentence from a bag of words. However, it is still challenging for neural models to handle large bags or the bags where some information is missing.
- Compared with humans, neural models show a better capability to organize sentences from a bag of words, especially in terms of large bags. However, the performance of humans is more robust to noisy words or missing words than neural models.

Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Science Foundation of China (NSFC No. 61876196) and the National Key Research and Development Program of China (No. 2020AAA0105200). Rui Yan is supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Adrià De Gispert, Marcus Tomalin, and Bill Byrne. 2014. Word ordering with phrase-based grammars. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 259–268.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252.

Eva Hasler, Felix Stahlberg, Marcus Tomalin, Adri de Gispert, and Bill Byrne. 2017. A comparison of neural models for word ordering. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 208–212.

Jing He and Hongyu Liang. 2011. Word-reordering for statistical machine translation using trigram language model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1288–1293.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

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

Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches out: Proc. ACL-04 Workshop*, pages 74–81.

Jiangming Liu and Yue Zhang. 2015. An empirical comparison between n-gram and syntactic language models for word ordering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 369–378.

Yijia Liu, Yue Zhang, Wanxiang Che, and Bing Qin. 2015. Transition-based syntactic linearization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 113–122.

Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. 2018. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, volume 2, page 3.

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.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

Allen Schmaltz, Alexander M Rush, and Stuart M Shieber. 2016. Word ordering without syntax. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 16, pages 3776–3784.

Cyrus Shaoul. 2010. The westbury lab wikipedia corpus. *Edmonton, AB: University of Alberta*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Songhua Xu, Shaohui Yang, and Francis Chi-Moon Lau. 2010. Keyword extraction and headline generation using novel word features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1461–1466.

Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. 2020. Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction. *International Journal of Computer Vision*, 128(1):53–73.

Yue Zhang, Graeme Blackwood, and Stephen Clark. 2012. Syntax-based word ordering incorporating a large-scale language model. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 736–746. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2011. Syntax-based grammaticality improvement using ccg and guided search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1147–1157. Association for Computational Linguistics.