Controlling the Focus of Pretrained Language Generation Models

Anonymous ACL submission

Abstract
The finetuning of pretrained transformer-based language generation models are typically conducted in an end-to-end manner, where the model learns to attend to relevant parts of the input by itself. However, there does not exist a mechanism to directly control the model’s focus. This work aims to develop a control mechanism by which a user can select spans of context as “highlights” for the model to focus on, and generate relevant output. To achieve this goal, we augment a pretrained model with trainable “focus vectors” that are directly applied to the model’s embeddings, while the model itself is kept fixed. These vectors, trained on automatic annotations derived from attribution methods, act as indicators for context importance. We test our approach on two core generation tasks: dialogue response generation and abstractive summarization. We also collect evaluation data where the highlight-generation pairs are annotated by humans. Our experiments show that the trained focus vectors are effective in steering the model to generate outputs that are relevant to user-selected highlights.

1 Introduction
Transformer-based models pretrained on large-scale text data have become the dominant paradigm for natural language generation (NLG) tasks (Roller et al., 2020; Lewis et al., 2019; Raffel et al., 2020). The attention module (Bahdanau et al., 2016; Vaswani et al., 2017), which aggregates information via a weighted average over word-level embeddings, plays a vital role in these models. The attention mechanism serves two major purposes: (1) It captures linguistic phenomena in the input (Clark et al., 2019; Kovaleva et al., 2019; Kobayashi et al., 2020); (2) It helps the model focus on relevant portions of the input (e.g., alignment in machine translation (Bahdanau et al., 2016) and abstractive summarization (Rush et al., 2015)).

The attention module is particularly useful as it does not require any explicit supervision: the model learns to attend to relevant parts of the input by itself through end-to-end training. However, this property makes it difficult to explicitly control the model’s focus. If the model happens to put focus on some span of context that the user thinks is not so important, we do not have a mechanism to correct it. This is especially sub-optimal in some NLG applications involving a relatively long input such as dialogue or summarization: focusing on different spans of the input could result in completely different generations (illustrated in Figure 1). It would be attractive to give the user an option to control the model’s focus.

In this work, we aim to develop a mechanism to steer the model to generate output relevant to some user-specified input spans (which we term as highlights).1 This goal, however, brings about significant challenges. For one, many popular NLG datasets are collected in an end-to-end manner, i.e., without annotations of which spans of input are most relevant to the reference target. It would also be ideal for the proposed approach to be compatible

---

1To avoid confusion, our goal is not about controlling the attention modules inside the model, instead, we care about the actual generation.
with existing pretrained transformer models, as re-
training such models is often costly.
In this work, we propose an focus vector frame-
work to address the challenges outlined above. Our
contributions are as follows:

- To control the model’s focus, we augment the
pretrained model with trainable focus vectors
which are directly applied to the encoder em-
bettinhs. The model itself is kept fixed, and
no further changes to the model architecture
is needed.

- The training of focus vectors does not require
additional annotations. We utilize attribution
methods to derive automatic highlight annota-
tions from existing end-to-end training data.

- For principled evaluation and future work
in this direction, we collect and release hu-
man evaluation data where the highlight-
generation pairs are annotated by humans.

- We test our method on two core NLG tasks:
dialogue response generation and abstractive
summarization. Experiments show that the
trained focus vectors are effective in steering
the model to generate a relevant output given
the selected highlights.

2 Model Formulation

We assume the target model is a standard pretrained
transformer encoder-decoder model (Vaswani et al.,
2017) that has already been finetuned on end-to-end
task-specific data (e.g., dialogue or summarization)
with the standard negative log-likelihood (NLL)
loss. Our goal is to establish a control mechanism
whereby the user can highlight several spans of
the input, and the model is supposed to generate
outputs relevant to the highlighted text. Crucially,
this mechanism should not change the base model,
in order to allow the user to default back to the
original model if desired.

We begin by establishing notation. We de-
note the end-to-end training data by \( \{x, y\} \), where
\( x = \{x_1, \ldots, x_n\} \) refers to the input token sequence,
and \( y \) refers to the corresponding reference target
token sequence. During evaluation, some spans of
the input \( x \) will be highlighted, and we use a binary
indicator \( c_i \) to indicate whether the \( i^{th} \) input token
is to be highlighted during generation. In this work
we only consider a set of complete sentences as a
valid highlight span. This design choice is mainly
for convenience during our human-annotated eval-
uation data collection, and our framework can readily
be generalized to phrase-level highlights.

Suppose the encoder model is composed of \( L \)
transformer layers. We denote the \( d \)-dimensional
output embedding of the \( i^{th} \) position on the \( l^{th} \)
encoder layer by \( h_i^l \). We use \( \{h_i^l\} \) to denote the
input embeddings. Each decoder layer performs
multi-head cross-attention on the outputs of the en-
coder, where the attention weight computation for
the \( h_i^l \) head on the \( l^{th} \) decoder layer is formulated
as below:

\[
\alpha_{i,j}^{h,l} = \text{softmax}_{i\in\{1,\ldots,n\}} \left( \frac{k(h_i^L) \cdot q_j^{h,l}}{\sqrt{d}} \right). \tag{1}
\]

Here \( k(\cdot) \) is a linear transform, and \( \alpha_{i,j}^{h,l} \) is the
attention weight assigned to encoder output \( h_i^l \), for
the \( j^{th} \) position decoder query vector \( q_j \). We use
\( P_M(y|x) \) to denote the probability assigned to \( y \)
given input \( x \) by the original target model. For
more details of the transformer encoder-decoder
architecture, we refer readers to Vaswani et al.
(2017).

Our proposed framework involves two stages.
We first obtain automatic highlight annotations us-
ing attribution methods. Then, these annotations
are used to train the focus vectors. In the next
section, we review the attribution methods.

2.1 Attribution Methods

Many popular NLG datasets are collected end-to-
end, i.e., without annotations of which spans of
input are most relevant to the reference target. To
obtain these annotations for focus vector training,
we make use of existing attribution methods.

Attribution methods (Baehrens et al., 2010; Si-
monyan et al., 2014; Shrikumar et al., 2017; Adep-
bayo et al., 2018; Sundararajan et al., 2017), also
known as saliency maps, attribute the prediction of
a (potentially black-box) model to its input features.
It thus fits our need to extract relevant spans in
the input given the reference target. Most saliency
methods are originally designed for image clas-
sification, where an importance score is assigned
for each dimension of the input feature. There-
fore, slight modifications (e.g., dot-product with
the word embeddings) are needed to apply them to
language data (Ding and Koehn, 2021; Denil et al.,
2014).

We implement and compare several popular at-
tribution methods, which compute the attribution

\[
\alpha_{i,j}^{h,l} = \text{softmax}_{i\in\{1,\ldots,n\}} \left( \frac{k(h_i^L) \cdot q_j^{h,l}}{\sqrt{d}} \right). \tag{1}
\]
score for a given sentence $S$ (denoting the set of
token indexes in the sentence) in the input $x$ for the
target $y$ and model $P_M$.

**Leave-one-out (LOO)** We replace the tokens in
$S$ by the $<$pad$>$ token, and compute the difference
in NLL:

$$A(S) = \log P_M(y|x) - \log P_M(y|x_{\text{S-padded}}). \quad (2)$$

LOO is also referred to as an occlusion-based method (Zeiler and Fergus, 2014; Li et al., 2016)
in the literature.

**Attention-weight** We sum up the attention
weights assigned to tokens in $S$ for all attention
heads across all decoder layers:

$$A(S) = \sum_{i} \sum_{j \in S} \alpha_i^{h,j}. \quad (3)$$

**Grad-norm** We sum the norm of gradient for the
input word embeddings in $S$:

$$A(S) = \sum_{i \in S} \|\nabla_h^i \log P_M(y|x)\|_2. \quad (4)$$

**Grad-input-product** Instead of taking vector
norm, we compute the dot-product between the
input embedding and its gradient:

$$A(S) = \sum_{i \in S} \left( \nabla_h^i \log P_M(y|x) \right) \cdot h_i^0. \quad (5)$$

While more sophisticated attribution method have
been proposed in the literature (Lei et al., 2016;
Sundararajan et al., 2017; Bastings et al., 2019),
we mainly experiment with the methods listed above
due to their simplicity and popularity. Attribution
methods have been used for interpreting black-box
models—applying them to derive labels that can
further be used to control the focus of a model
has to our knowledge not been explored before.

Which attribution method best reflects the
model’s inner working is still an active research
area (Ding and Koehn, 2021; Adebayo et al., 2018).
The present work is primarily concerned with how
well the attribution scores align with human-
annotated highlights. In our experiments, we find
that leave-one-out (LOO) has the best correlation
on the human-annotated development set (Table
1, details given in Section 3). We therefore adopt
LOO to derive the automatic highlight annotations.

More specifically, for the input-output pairs in
the training set, we sort the LOO attribution scores
of the sentences in the input from large to small,
and mark the tokens in the first few sentences (the
exact number varies by task) as highlights. We
denote the highlight labels obtained from this au-
tomatic procedure by a binary indicator variables
$\mathbf{c}^{\text{attr}} = \{c_1^{\text{attr}}, \ldots, c_n^{\text{attr}}\}$, which will be used to train
the focus vectors.

### 2.2 Focus Vectors

To control the model’s focus, we introduce a set of
d-dimensional vectors $\theta$, named focus vectors.
They are designed to act as indicators for the
model, designating which parts of the input to focus
on. We now assume the training set contains
$\{x, c^{\text{attr}}, y\}$ triples, where $c^{\text{attr}}$ is obtained from the
attribution method from the previous section. Focus
vectors modify the forward pass of the encoder
model by applying a simple transformation $f$ on
the output embeddings of each layer (including the
input layer):

$$f(h_i^t) = \begin{cases} h_i^t \odot \theta_{\text{scale-focus}} + \theta_{\text{bias-focus}}, & \text{if } c_i^{\text{attr}} = 1 \\ h_i^t \odot \theta_{\text{scale-nonfocus}} + \theta_{\text{bias-nonfocus}}, & \text{if } c_i^{\text{attr}} = 0 \end{cases}. \quad (6)$$

We provide an illustration in Figure 2. The total
number of parameters introduced by the focus vec-
tors is therefore $4 \times (L + 1) \times d$, which is negligible
in comparison to the large number of parameters
of the fixed transformer model. We note that as
the focus vectors operate directly on the encoder
embeddings, it does not require an explicit atten-
tion module to exist in the model and is therefore
applicable to non-attentional architectures such as
LSTMs (Huang et al., 2015).

We train the focus vectors using the standard
NLL loss with stochastic gradient descent (SGD):

$$\mathcal{L}(x, y; c^{\text{attr}}; \theta) = -\log P_{\text{focus}}(y|x; c^{\text{attr}}), \quad (7)$$

where $P_{\text{focus}}$ denotes the distribution over
the output after the focus vectors are applied. We
We test our method on two NLG tasks: dialogue and summarization. To save space, we defer details and statistics of the datasets to Appendix A. The participants are required to get to know each other by chatting naturally. Each of them is given a persona: several pieces of personal information such as "I major in Computer Science", serving as background information. The participants are required to reflect their assigned persona in the conversation. For summarization, we adopt the CNN/Dailymail dataset (Hermann et al., 2015; Nallapati et al., 2016), which is a standard dataset for end-to-end abstractive summarization. To save space, we defer details and statistics of the datasets to Appendix A.

Both PersonaChat and CNN/Dailymail are created end-to-end and do not contain annotated highlight spans. For principled evaluation, we utilize the Amazon Mechanical Turk (AMT) platform to collect evaluation sets where the highlight-generation pairs are annotated by humans.

For PersonaChat, each turker\(^2\) is shown a dialogue history and the corresponding persona of the speaker. The dialogue history is randomly selected from the original test set of PersonaChat. Then the turker is required to choose 1-3 sentences as highlights (for example, one sentence in persona, and one sentence in dialogue history), and write down a dialogue response that not only continues the current dialogue, but also is relevant to the chosen highlights. Finally, we ask the turker to repeat the above process, but select a different set of highlights and provide another response. After a few preliminary trials and modifications to our instructions / rewards, we find that turkers comply nicely with our instructions and provide high-quality highlight-response pairs.

For CNN/Dailymail however, we first found that turkers had difficulty writing a high-quality summary for a given news article, with many turkers giving random responses even after we increased the reward. This is perhaps unsurprising given that writing a good summary is challenging and the reference summaries are written by experts. After a few disappointing iterations, we turn to a compromise: we directly provide the turkers with the reference summary, and only ask them to select 2-5 relevant sentences in the article. This simplifies the task, and we are able to collect high-quality labels. This compromise is not ideal, as it reverses the order of highlighting and generation. However, we find that in most cases, the reference summaries in CNN/Dailymail are well covered by several “key” sentences in the article, which are highlighted by the turkers. Therefore, we believe this compromise does not hurt the soundness of our evaluation.

In order to ensure high data quality for both dialogue and summarization, we design a qualification test that turkers need to pass before conducting the actual tasks. Several automatic checks and a minimal time limit are added in the scripts to prevent trivial answers. We also manually monitor the incoming submissions, and ban misbehaving turkers and discard their submissions. More details about our AMT setup are provided in Appendix B.

Our final collected datasets include 3,902 highlight-generation pairs for PersonaChat, and 4,159 pairs for CNN/Dailymail. They are randomly split 50/50 into dev/test sets. We include a number

\(^2\)We recruit turkers located in the U.S.
of samples of our collected data in the supplementary materials. Our code and the collected dataset will be released in the public version of this manuscript. We hope that this evaluation data could facilitate future research in this direction.

Comparison of Attribution Methods We use the collected highlight-generation pairs in the dev set to compare which attribution method aligns best with human-annotated highlights. In particular, we compute the top-1 precision of the sentence ranked highest by the attribution method. The result is shown in Table 1. We find that for both PersonaChat and CNN/Dailymail, LOO has the best alignment. We therefore use LOO to obtain automatic annotations for focus vector training. Interestingly, we observe low alignment between attention weight-derived attribution scores and human judgment, which indicates that controlling model generations via intervening on the attention distributions may not optimal. Finally, we note that this result does not mean LOO is the “best” attribution method, as attribution method is supposed to reflect the model’s inner working, instead of a human’s.

4 Experiments

4.1 Experiment Setting and Baselines

We use Blenderbot (Roller et al., 2020) as the base model for PersonaChat and BART (Lewis et al., 2019) for CNN/Dailymail, both of which are standard encoder-decoder transformer models. Our code is based on the transformers library (Wolf et al., 2020). We load the pretrained weights from facebook/blenderbot-400M-distill and facebook/bart-base. Blenderbot has 2 encoder layers and 12 decoder layers, while BART has 6 encoder layers and 12 decoder layers. To help Blenderbot cope with long dialogue context in PersonaChat, we extend its maximum position embedding index from 128 to 256. We use beam-search for decoding, where we follow the recommended configuration (Roller et al., 2020; Lewis et al., 2019) and use a beam size of 10 for Blenderbot and a beam size of 4 for BART.

For both tasks, we first finetune the base model on the original training set in the standard end-to-end manner. The model is then fixed and used to obtain automatic labels \( c^{\text{attr}} \) with the LOO attribution method on the same training set. For each training sample, we select the top-\( k \) sentences in the input ranked by LOO. Since we do not know the best value for \( k \), we set it to be a random number from 1 to 3 for PersonaChat, and from 2 to 5 for CNN/Dailymail.

While the highlight labels in the training set used to train focus vectors are derived automatically, we use the human-labeled dev set for hyper-parameter tuning. This is to facilitate fair comparison with other baseline approaches which also utilize the human-labeled dev set. In our ablation study, we will show that this dependence on human-labeled dev set is not crucial for our approach to achieve strong performance. We perform a grid search over learning rate with \( \{1, 3, 5\} \times \{1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}\} \). The Adam optimizer (Kingma and Ba, 2014) is used with \( \beta_1 = 0.9, \beta_2 = 0.999 \), and a L2 decay weight of 0.01. For both tasks, we set the mini-batch size to be 16.

We compare the proposed focus-vector approach with several baselines:

Vanilla: The vanilla model, without any modification in both the model and the input.

Padding: One trivial way to control the model’s focus is to replace all input by the \(<\text{pad}>\) token, except the spans highlighted by the user. However, we find that this direct padding during evaluation results in drastically worse perplexity. To alleviate this problem, we randomly pad a portion of sentences in the input during the standard end-to-end finetuning, to make the model aware that only partial input would be provided.

Keyword-control: Keyword-based prompts (Fan et al., 2017; He et al., 2020) has been a popular approach for controllable text generation. We adapt this idea to our focus-control setting. During model finetuning, we prepend key-phrases extracted from the reference target sequence to the original input. We utilize Yake (Campos et al., 2020), which is an unsupervised keyword extraction method. During evaluation, we extract and prepend key-phrases extracted from the highlighted sentences.

Attention-offset: As a direct way to control the model’s attention, we add a positive scalar offset \( s^{\text{offset}} \) to the cross-attention heads before the softmax operation (Equation 1), for the highlighted spans. A similar technique has been used in Dong et al. (2021) to modulate the attention distribution to tackle neural text degeneration problems (Holtzman et al., 2019). This approach modifies the at-
### 4.2 Results and Analysis

During evaluation, human-annotated highlights are fed to the model. In addition to perplexity, we evaluate the generations from different approaches using two popular NLG metrics: ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2019).

We show the main results in Table 2. As expected, the padding baseline has poor performance, as a large portion of input is masked out. Comparing to various baselines, focus-vector obtains significantly improved ROUGE and BERTScore on both tasks. This validates the motivation of this work: focus-vec is effective in steering the model’s focus, which leads towards the desired generation. For CNN/Dailymail, the perplexity of focus-vector is close to the vanilla model even though there is a large difference in ROUGE. We believe this is due to the constrained nature of the summarization task and how perplexity is computed: once the model observes the first few tokens, it is easy to figure out what the current highlight is. The other two metrics, on the other hand, are based on the actual generation, and therefore do not have this issue.

The performance of keyword-control, although better than the vanilla model, is inferior to attention-offset and focus-vector. We surmise this is due to the following two weaknesses: First, key-phrases can not fully represent the highlighted span. Second, there is a discrepancy of where the key-phrases are extracted between training and evaluation.

The performance gap (in ROUGE/BERTScore) between focus-vector and attention-offset is larger on the CNN/Dailymail dataset. We believe this is because the BART model has a deeper encoder than the Blenderbot model. As the encoder grows deeper, the embeddings become more “contextualized” and its identifiability (Brunner et al., 2020) degrades. And since the decoder attends to the last layer of the encoder, this direct manipulation of attention weights could be ineffective with deep encoders.

Table 3 shows generation samples from different focus-control approaches for PersonaChat. Spans of the generation that are relevant to the highlighted persona are marked in red. Comparing to the generation from the vanilla model, the generations from both attention-offset and focus-vector are highly relevant to the respective highlighted persona. One generation from att-offset is a little erratic (“I am petro, my dog”), which may be due to the inflexibility of att-offset.

We defer the generation examples for CNN/Dailymail to Table 6 and Table 7 (Appendix D) due to space constraints. We observe that the generation from focus-vector is more focused on the highlighted inputs. On the other hand, attention-offset’s generation still remains similar to the vanilla model.

In Figure 3, we study how the outputs of attribution methods (attention-weight and grad-norm) change with different approaches (vanilla, focus-vector and attention-offset) for the CNN/Dailymail example (Table 6). Note that in this analysis, for the attribution methods we set the target $y$ to be the decoded output from the respective modeling.
I have a dog named pedro. I like to eat muffins. I am doing well. How about you? What do you do for a living? I am in healthcare. I am doing well. I am petro, my dog, and I am doing well. I am petro, my dog, and I am good. Just got back from walking my dog. Do you have any pets? Hi! I am doing well. How about you? What do you do for a living? I am in healthcare.

Table 3: Generation samples from different attention control approaches for PersonaChat. The inputs on both sides are identical, except a different piece of persona is highlighted (marked in bold), which leads to different generations for approaches with control. Another example is provided in Table 5 (Appendix D).

<table>
<thead>
<tr>
<th>persona:</th>
<th>I have a dog named pedro. I like to eat muffins. I am five feet tall. I work in the healthcare industry.</th>
</tr>
</thead>
<tbody>
<tr>
<td>context:</td>
<td>[P-1]: Hello, how are you today? [P-2]: [TO ANSWER]</td>
</tr>
<tr>
<td>reference:</td>
<td>Scared, i’m looking for my dog, his name is pedro. Have you seen him?</td>
</tr>
<tr>
<td>vanilla:</td>
<td>Hi! I am doing well, how about yourself? What do you do for a living?</td>
</tr>
<tr>
<td>keyword:</td>
<td>I am doing well, how about yourself? What do you like to do for fun?</td>
</tr>
<tr>
<td>att-offset:</td>
<td>I am doing well, and you? I am petro, my dog, and you?</td>
</tr>
<tr>
<td>focus-vec:</td>
<td>I am good. Just got back from walking my dog. Do you have any pets?</td>
</tr>
</tbody>
</table>

Table 4: Performance of different variants of focus-vector trained on CNN/Dailymail. all-layer* refers to our proposed modelling (also reported in Table 2).

<table>
<thead>
<tr>
<th>Model</th>
<th>CNN/Dailymail</th>
<th>PPL</th>
<th>ROUGE-1/2/L</th>
<th>BERTScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-layer*</td>
<td>4.45</td>
<td>45.92/23.03/32.98</td>
<td>89.78</td>
<td></td>
</tr>
<tr>
<td>only first layer</td>
<td>4.48</td>
<td>45.67/22.63/32.45</td>
<td>89.59</td>
<td></td>
</tr>
<tr>
<td>only last layer</td>
<td>4.48</td>
<td>45.66/22.84/32.69</td>
<td>89.69</td>
<td></td>
</tr>
<tr>
<td>plus model finetune</td>
<td>4.49</td>
<td>46.65/23.54/33.30</td>
<td>89.82</td>
<td></td>
</tr>
</tbody>
</table>

5 Related Work

Our proposed focus-vector framework is closely related to the research topics of controllable text generation, LM adaptation, and attention/attribution analysis, which we review below.

Controllable Text Generation

Prior work on controllable summarization introduced various types of control mechanisms. Fan et al. (2017); Saito et al. (2020) extract entity, keyword or length, as additional supervision during training. Gehrmann et al. (2018) trains a token-level content selection module, where the supervision is by aligning the summaries to the documents. (Song et al., 2021) proposes a two-staged generation strategy and Goyal et al. (2021) incorporates multiple decoders into a transformer framework. Some recent work (He et al., 2020; Dou et al., 2020) uses prompts to control the generation. Lexically constrained decoding (Post and Vilar, 2018) has also been used to enforce certain key phrases to be included in the summary (Mao et al., 2020).

Existing work on controllable dialogue response generation include using conditional variational au-
to encoders (Zhao et al., 2017; Li et al., 2020), and incorporating external knowledge into the conversational agent using knowledge graphs (Cui et al., 2021; Moon et al., 2019), unstructured documents (Kim et al., 2020), or dialogue context (Zhao et al., 2020).

In open-ended language generation, a series of approaches have been proposed to control for some attribute (e.g., topic) of the generation (Keskar et al., 2019; Dathathri et al., 2020; Krause et al., 2020; Yang and Klein, 2021). Some of these studies utilize a trained classifier to guide the generative model towards the desired attribute.

**LM Adaptation** Our proposed focus vector framework is also inspired by a series of recent works on prompting or light-weight LM adaptation. Li and Liang (2021), followed by Lester et al. (2021) and Zhong et al. (2021), propose prefix tuning, where continuous task-specific input vectors are tuned to adapt the pretrained LM to a downstream task with supervised data, and the model is kept fixed.

There is also a line of works on adapter-tuning, which insert and finetune task-specific layers (adapters) between each layer of the pretrained LM (Houlsby et al., 2019; Lin et al., 2020; Pfeiffer et al., 2021). More recently, Guo et al. (2021) and Ben-Zaken et al. (2020) propose to finetune only a small subset of a pretrained model’s parameters, and achieves strong performance on the GLUE benchmark (Wang et al., 2018).

**Attention Analysis and Attribution Methods** Due to the ubiquity of the attention module in current NLP models, various work has studied how the module captures linguistic phenomena in the input (Clark et al., 2019; Kovaleva et al., 2019; Kobayashi et al., 2020). It has also been used as a tool to interpret the model’s predictions (Wang et al., 2016; Lee et al., 2017; Ghaeini et al., 2018).

Recently, there have been a series of studies discussing the use of attention weights for interpretability (Jain and Wallace, 2019; Wiegrefe and Pinter, 2019; Bastings and Filipppova, 2020; Serrano and Smith, 2019), and it has been argued that attribution methods are a better choice to explain the model’s predictions. The poor alignment performance of attention weights that we get in Table 1, on some level, are in agreement with that argument. Our work is also related to the line of work on interpreting black box models through rationales (Lei et al., 2016; Bastings et al., 2019), which are typically (discrete) subsets of the input that are used to predict the output. Finally, several recent works (Xu and Durrett, 2021; Ding and Koehn, 2021) have compared different attribution methods for interpreting NLP models.

In comparison to the aforementioned works, our major innovations are two fold: (1) Our goal is to control the focus of pretrained models, and thereby steer the model’s generation, and our proposed focus vectors are compatible with the standard transformer architecture; (2) We utilize attribution methods to get automatic highlight labels for training focus vectors. Therefore, our framework can be applied to a wide range of NLG applications.

**6 Conclusion**

In this work we propose the focus vector framework as a light-weight solution to control the focus of pretrained transformer models. It has two major advantages: (1) Focus vectors act as simple transformations to the embeddings in the encoder, and the transformer model is kept fixed; (2) Attribution methods are utilized to get automatic highlight labels for training focus vectors.

We test our approach on two tasks: dialogue response generation, and abstractive summarization. For evaluation, we collect data where the highlight-generation pairs are annotated by humans. Experiments show that the trained focus vectors are effective in steering the model to generate output text that is relevant to the specified highlights.
References


Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In Com-


Appendices

A End-to-end Datasets

The PersonaChat dataset contains 8,939 dialogues for training, 1,000 for validation, and 968 for test. For each turn in the dialogue, we concatenate the persona of the speaker and the dialogue history as input, and train the base model to generate the current utterance. In some cases, the dialogue history is long and exceeds the input limit of the model, in which case we truncate the dialogue at the sentence level. The average number of sentences is around 11 after truncation.

The CNN/Dailymail dataset contains 287,113 training examples, and 13,368 / 11,490 examples for validation / test. We apply the same truncation strategy as PersonaChat during preprocessing. The processed articles have an average length of 748 tokens, and the reference summaries have an average length of 67 tokens.

B Human-annotated Evaluation Data Collection

To improve the quality of collected dataset, we design a qualification test, which the turkers need to pass before they can work on real assignments. The test is designed to help turkers understand our task better. For PersonaChat, we give turkers two dialogue samples with pre-selected highlights, and ask them to choose the appropriate response that not only continues the dialogue, but also is relevant to the highlights. For CNN/Dailymail, the turkers are shown two example articles and the corresponding reference summaries. We have already picked some highlights in the article, but there is one highlight missing. And the turker is required to pick the missing highlight. The interface for the PersonaChat qualification test is shown in Figure 5.

We also add multiple checks in our script to prevent trivial answers. We ban trivial copy&paste from the given context. A time check is added that requires turker to spend at least 60 seconds on a single HIT. For the two assignments in PersonaChat, we add a content check that prevents duplicate highlights or response. We show our interface for PersonaChat in Figure 6. Despite these checks and the qualification tests, there still exist a small number of misbehaving turkers who attempt to cheat. Therefore we also manually monitor the incoming submissions, and ban misbehaving turkers and filter out their submissions.

More examples of our interface and instructions can be found in our uploaded data samples.

C Implementation Details

For the attention-offset baseline, we tune the offset $\sigma$ in a fine-grained manner, on the human-annotated dev set. We first set a relatively large max value (100) and get 20 evenly spaced numbers inside the interval $(0, 100)$. Then we calculate model PPL on the dev set with $\sigma$ set to these different offsets. Then we do another search in the interval that has lowest PPL. We repeat this iteration multiple times, and stops when PPL change is smaller than $10^{-3}$. The final tuned value for Blenderbot is around 3.02, and around 0.17 for BART.

D Auxiliary Results and Examples

In Figure 4, we provide a simple visualization of the trained focus vector on first encoder layer of the BART model. Figure 5: An example of our AMT qualification test for PersonaChat. We have chosen the highlights in the context, and the turker is supposed to choose a response that not only continues the dialogue, but also is relevant to the highlights.
figure easy to grasp, we randomly sample 50 di-

dimensions (out of 768) of trained focus-vector pa-

rameters. In the figure only the trained parameters

for the first encoder layer is shown, and we find

that the patterns for the other layers are similar.

We first find that the learned scaling parameters

are close to 1 and the bias parameters are close to

zero. This implies that the transformation defined

by the focus-vec is not drastic, and a relatively

small tweak could be enough to steer the model’s

focus.

An interesting pattern can be observed for the

scaling vectors. $\theta_{\text{scale-nonfocus}}$ tends to down-scale

the embedding, comparing to $\theta_{\text{scale-focus}}$. This

matches our intuition that the embeddings on the

non-highlighted positions should be down-played.
| persona: | My parents did not want me. It’s a dead end job so i am looking for something different. I was raised by my uncle and aunt. | My parents did not want me. It’s a dead end job so i am looking for something different. I was raised by my uncle and aunt. |
| context: | [P-1]: Hello how are you doing? [P-2]: Pretty good. how are you? [P-1]: I am making dinner getting ready for bed. [P-2]: I just got off work. [P-1]: I am looking for something different to do. [P-2]: [TO ANSWER] | [P-1]: Hello how are you doing? [P-2]: Pretty good. how are you? [P-1]: I am making dinner getting ready for bed. [P-2]: I just got off work. [P-1]: I am looking for something different to do. [P-2]: [TO ANSWER] |
| reference: | Do you know anywhere that’s hiring work from home? I’m looking for a new job. | I grow up with my aunt and we always had scripture study after dinner, you can try it. |
| vanilla: | My parents wanted me to be a doctor like them, but i do not want to do that. | My parents wanted me to be a doctor like them, but i do not want to do that. |
| keyword: | What do you do for work? I am a cashier at a fast food restaurant. | What do you do for work? I am a cashier at a fast food restaurant. |
| att-offset: | What do you do for a living? I have been looking for a new job myself. | Do you have any relatives you can ask for help with? my family was always there for me. |
| focus-vec: | What do you do for a living? I am unemployed right now looking for a job. | What do you do for a living? I work with my aunt at her law firm. |

| article: | England World Cup hopeful Elliot Daly has been named the Aviva Premiership’s player of the month for March. The uncapped Wasps centre, 22, scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Equally comfortable at full back, Daly’s form at centre has him pushing incumbent England No 13 Jonathan Joseph hard for his place. Elliot Daly is handed his player of the month award by Wasps director of rugby Dai Young. He looks certain to be included in Stuart Lancaster’s England’s World Cup training squad named next month. Wasps director of rugby Dai Young said: ‘If England are going to go out and win the Rugby World Cup, he’s the type of player you’ve got to have in the team so I’d certainly like to think he’s pushing for a place in the squad.’ BT Sport analyst Austin Healey added: ‘For a good couple of years now us rugby watchers have known of the potential of Elliot Daly.’ |
| reference: | Elliot Daly was in fine form at outside centre for Wasps in March. Daly, 22, has not yet been capped by England. Wasps head coach Dai Young says England must pick him for World Cup. | England World Cup hopeful Elliot Daly has been named Aviva Premiership’s player of the month for March. The 22-year-old scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Daly’s form at centre has him pushing incumbent England No 13 Jonathan Joseph hard for his place. |
| vanilla: | England World Cup hopeful Elliot Daly has been named Aviva Premiership’s player of the month for March. The 22-year-old scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Daly looks certain to be included in Stuart Lancaster’s England’s World Cup training squad. | England World Cup hopeful Elliot Daly has been named Aviva Premiership’s player of the month for March. The 22-year-old scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Daly’s form at centre has him pushing incumbent England No 13 Jonathan Joseph hard for his place. |
| keyword: | England World Cup hopeful Elliot Daly has been named Aviva Premiership’s player of the month for March. The uncapped Wasps centre scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Daly’s form at centre has him pushing incumbent England No 13 Jonathan Joseph hard for his place. Wasps director of rugby Dai Young said: ‘If England are going to go out and win the Rugby World Cup, he’s the type of player you’ve got to have in the team.’ | England World Cup hopeful Elliot Daly has been named Aviva Premiership’s player of the month for March. The 22-year-old scored two tries in March as the Coventry-based club maintained on course for a Champions Cup spot next season. Daly’s form at centre has him pushing incumbent England No 13 Jonathan Joseph hard for his place. |

Table 5: Generation samples from different attention control approaches for PersonaChat. The inputs on both sides are identical, except a different piece of persona is highlighted, which leads to different generations from att-offset and focus-vec.

Table 6: Generation samples of different attention control approaches on CNN/Dailymail dataset. The highlighted sentences are marked in bold. The span marked by red in the reference summary is captured by the generation from focus-vec, but not by att-offset. The sentence number marked in the input corresponds the sentence index in Figure 3.
Kabul, Afghanistan (CNN) A suicide bomber detonated his explosives near a group of protesters in eastern Afghanistan on Thursday, killing 17 people and wounding dozens more, police said. An Afghan lawmaker taking part in the protests in the city of Khost was among the 64 people wounded, said Faizullah Ghairat, the provincial police chief. Taliban spokesman Zabiullah Mujahid denied his group was responsible for the attack. No other organization has so far claimed responsibility. The suicide attack hit the group around 10 a.m. local time, police said.

| article: | Kabul, Afghanistan (CNN) A suicide bomber detonated his explosives near a group of protesters in eastern Afghanistan on Thursday, killing 17 people and wounding dozens more, police said. An Afghan lawmaker taking part in the protests in the city of Khost was among the 64 people wounded, said Faizullah Ghairat, the provincial police chief. Taliban spokesman Zabiullah Mujahid denied his group was responsible for the attack. No other organization has so far claimed responsibility. The suicide attack hit the group around 10 a.m. local time, police said. |
| reference: | An Afghan lawmaker is among 64 people wounded in the attack, police say. Taliban spokesman denies his group was responsible for the attack. |
| vanilla: | A suicide bomber detonates his explosives near a group of protesters, killing 17 people and wounding dozens more. The Taliban spokesman denies his group was responsible for the attack. No other organization has so far claimed responsibility for the attack. The suicide attack hit the group around 10 a.m. local time. |
| keyword: | Taliban spokesman Zabiullah Mujahid denies his group was responsible. No other organization has so far claimed responsibility. The suicide attack hit the group around 10 a.m. local time. |
| att-offset: | A suicide bomber detonates his explosives near a group of protesters, killing 17 people and wounding dozens more. The Taliban spokesman denies his group was responsible for the attack. No other organization has so far claimed responsibility for the attack. The suicide attack hit the group around 10 a.m. local time. |
| focus-vec: | A suicide bomber detonates his explosives near a group of protesters, killing 17 people. An Afghan lawmaker is among the 64 people wounded, police say. Taliban spokesman Zabiullah Mujahid denies his group was responsible for the attack. No other organization has so far claimed responsibility. |

Table 7: Generation samples of different attention control approaches on CNN/Dailymail dataset. The span marked by red in the reference summary is captured by the generation from focus-vec, but not by att-offset.