How do Large Language Models Learn In-Context? Query and Key Matrices of In-Context Heads are Two Towers for Metric Learning

Zeping Yu **Sophia Ananiadou**
Department of Computer Science, National Centre for Text Mining
The University of Manchester
{zeping.yu@postgrad. sophia.ananiadou@}manchester.ac.uk

Abstract

We investigate the mechanism of in-context learning (ICL) on sentence classification tasks with semantically-unrelated labels ("foo"/"bar"). We find intervening in only 1% heads (named "in-context heads") significantly affects ICL accuracy from 87.6% to 24.4%. To understand this phenomenon, we analyze the value-output vectors in these heads and discover that the vectors at each label position contain substantial information about the corresponding labels. Furthermore, we observe that the prediction shift from "foo" to "bar" is due to the respective reduction and increase in these heads' attention scores at "foo" and "bar" positions. Therefore, we propose a hypothesis for ICL: in in-context heads, the value-output matrices extract label features, while the query-key matrices compute the similarity between the features at the last position and those at each label position. The query and key matrices can be considered as two towers that learn the similarity metric between the last position's features and each demonstration at label positions. Using this hypothesis, we explain the majority label bias and recency bias in ICL and propose two methods to reduce these biases by 22% and 17%, respectively.

1 Introduction

In-context learning (ICL) is an emergent ability (Wei et al., 2022a) of large language models (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023). By using some demonstration-label pairs as prompts, ICL performs well without updating parameters on many tasks, such as machine translation (Sia and Duh, 2023), complexity reasoning (Li et al., 2023a), compositional generalization (Zhou et al., 2022) and information extraction (He et al., 2023).

Because the mechanism of ICL remains unclear, many studies focus on understanding how ICL works. Pan et al. (2023) find that ICL can be disentangled into task recognition (TR) and task learning (TL). TR does not rely on the demonstration-label mappings because the roles of demonstrations and labels are helping the model know "what is the task". In this situation, the model have similar predictions when the mappings are wrong (Min et al., 2022), because the predictions are based on pre-trained priors. On the other hand, TL relies on the demonstration-label mappings because the semantic priors are removed. For example, in an ICL sentiment classification task, if the labels are "positive/negative", the task is TR. If the labels are "foo/bar", the task is TL because the labels are semantically-unrelated (Wei et al., 2023). Wang et al. (2023) analyze the information flow by averaging all attention heads and find the label words are anchors to merge the semantic information of corresponding demonstrations in shallow layers, and information is extracted from label words to the final prediction in deep layers.

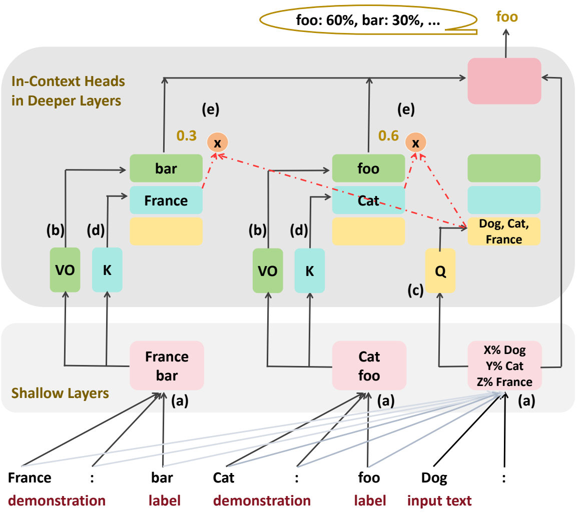

Figure 1: Hypothesis of ICL mechanism. (a) Shallow layers merge features into label positions and last position. In in-context heads, (b) value-output matrix VO extracts label information. (c) Query matrix Q and (d) key matrix K compute the (e) similarity scores between last position and each demonstration, deciding how much label information is transferred into the last token.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3281–3292
November 12-16, 2024 ©2024 Association for Computational Linguistics

Although these studies are important for understanding ICL, the exact mechanism of ICL remains a mystery for several reasons. Firstly, the information flow is typically observed as an average across each head, but understanding ICL requires exploring the precise importance of each head. Secondly, each head has a query matrix, key matrix, value matrix, and output matrix; it is essential to study the role of each matrix in detail. Lastly, ICL is plagued by issues such as majority label bias and recency bias, and how to explain and mitigate these biases has not yet been thoroughly investigated.

In this paper, we address these issues by identifying important heads for ICL and studying the roles of each matrix within these heads. Using two methods, we identify 12 important heads (named in-context heads) that significantly affect ICL accuracy across five datasets, reducing it from 87.6% to 24.4% on average. Intervening in 6 heads (fooheads) decreases the probabilities of "foo", while intervening in the other 6 heads (barheads) reduces the probabilities of "bar". To explore the reason of this phenomenon, we study these heads' value-output vectors computing by value-output matrices, and find that the vectors on label positions contain much information about the corresponding labels. Moreover, we observe the attention scores in the in-context heads when predictions shift from "foo" to "bar", and find that the attention scores at "foo" positions decrease, while the attention scores at "bar" positions increase. Based on these observations, we propose a hypothesis for ICL, as shown in Figure 1: in in-context heads, value-output matrices extract label information ("foo"/"bar") from corresponding labels, and query-key matrices compute the similarity between the last position and each label position. The query and key matrices can be regarded as two towers for learning the similarity between the features at last position and each demonstration at label positions. The greater the similarity, the higher the probability of the corresponding label.

Based on this hypothesis, we explore the reason why ICL has majority label bias (Zhao et al., 2021) and recency bias (Lu et al., 2021). The existing of majority label bias matches our hypothesis: query and key matrices compute the attention weights between the last position and each demonstration, so the sum of one label's attention weights is larger when this label is related to more demonstrations. About recency bias, we hypothesize that it is caused by the influence of positional embedding during attention score computation in both shallow and deep layers. Based on our analysis, we propose two methods for reducing these biases. For majority label bias, we increase the attention weight of the imbalanced label's position in in-context heads, and the majority label bias reduces 22%. For recency bias, we remove the affect of position embedding in in-context heads, and the recency bias reduces 17%. Our code and data will be released on `https://github.com/zepingyu0512/in-context-mechanism`.

2 Related Work

2.1 Understanding ICL

Many studies have explored the mystery of ICL. Min et al. (2022) find that randomly replacing the ground truth labels does not hurt performance much. Wei et al. (2023) argue the reason of this phenomenon is the model can rely on semantic priors. Therefore, they study semantically-unrelated label ICL by transferring the labels into "foo" and "bar" and find that the performance is related to the demonstration-label mapping. Pan et al. (2023), disentangle ICL into task recognition (TR) and task learning (TL) to explain this phenomenon. Chan et al. (2022) demonstrate that the ICL ability is obtained when training data have enough rare classes. Liu et al. (2021) argue that selecting the closest neighbors as demonstrations can enhance ICL ability. Gonen et al. (2022) propose choose low perplexity demonstrations to increase the performance of ICL. Dong et al. (2022) conclude these methods in a survey for ICL. Wang et al. (2023) find the label words are anchors to extract demonstrations in shallow layers, and the last position extracts information from label words in deep layers.

Some studies try to explain ICL theoretically. Xie et al. (2021) argue that ICL ability is gained when the pretraining distribution is a mixture of HMMs, and they explain ICL as implicit Bayesian inference. Garg et al. (2022) prove that transformers can learn linear functions by ICL. Akyürek et al. (2022) find transformers can learn linear regression functions and hypothesize that ICL can implement standard learning algorithms implicitly. Li et al. (2023b) explore the softmax regression and find that attention-only transformers are similar with gradient descent models. Von Oswald et al. (2023) and Dai et al. (2022) regard ICL as meta-learning and argue that ICL does gradient descent implicitly.

2.2 Mechanistic Interpretability

The goal of mechanistic interpretility (Olah, 2022; Nanda et al., 2023) is to reverse engineer the circuits from inputs to outputs. One common method is to apply gradient-based methods (Sundararajan et al., 2017; Kindermans et al., 2019) or causal tracing methods (Pearl, 2001; Vig et al., 2020; Meng et al., 2022) to analyze the importance of different attention heads and hidden states. Olsson et al. (2022) find that induction heads in attention layers are helpful for copying words from the input sequence (e.g. [X][Y]...[X] -> [Y]). Wang et al. (2022) interpret the circuits on indirect object identification task in GPT2. Hanna et al. (2023) studies how GPT2 computes greater-than by constructing a computational graph of head node and MLP node.

Another common method for mechanistic interpretability is the logit lens (Nostalgebraist, 2020), whose idea is to analyze the hidden vectors in unembedding space (also named vocabulary space). Many studies have found that the parameters in transformers are interpretable when projecting into vocabulary space (Elhage et al., 2021; Geva et al., 2022; Dar et al., 2022).

3 Hypothesis for ICL Mechanism

Our hypothesis is motivated by a case study in Section 3.1. We find that ICL performance can be affected much by only 1% heads, where some can enhance the probabilities for "foo" and others for "bar" (Section 3.2). To understand why this happens, we analyze the value-output vectors and attention scores in Section 3.3 and find that value-output matrices extract the label information and attention scores computed by query-key matrices control the label information flow. At last, we discuss our hypothesis for ICL in Section 3.4.

3.1 Hypothesis Motivated by Case Study

Our hypothesis and analysis is motivated by a case study in GPT2-large (Radford et al., 2019). We design a simple ICL case for word classification: **"love : bar like : bar eight : foo two : foo one :"**, where the model's prediction is **"foo"**. In this case, "foo" is the semantic-unrelated label for "number" and "bar" is for "sentiment". We propose a locate-and-project method for case study: we first locate the most important heads using the method discussed in Section 3.2, then project the vectors on label and last positions into vocabulary space by multiplying each vector v and the unembedding matrix E_u, following Dar et al. (2022):

$$D_v = softmax(E_u\, v) \tag{1}$$

Top tokens of value-output vectors and key vectors at label positions and query vector at last position in a_{22}^{0} (layer22, head0) are shown in Table 1.

position	top words in vocabulary space
2-value	**BAR**, Barron, Barrett, Band, Bray, **Bars**, Baron, **Bar**, Bay, Boyd
5-value	**BAR**, Barron, Barrett, Baron, **Bar**, Band, Barbie, Barbar, Bard
8-value	**foo**, **Foo**, FO, fo, Foley, Fresno, FDR, fascists
11-value	**foo**, **Foo**, fo, FO, fascists, FDR, Foley, Goo, fascists
2-key	**kisses**, **goddess**, **love**, **charms**, idol, stress, nobles, **happiness**
5-key	style, oriented, +++, **like**, indo, height, Lover, xual, dont, foo
8-key	foo, mc, blah, happ, avg, french, omega, prod, english, google, height, neigh
11-key	foo, mc, infinity, omega, **three**, **two**, repeat, **twelve**, 666, **Three**, **thirds**, **five**, **sixteen**
13-query	**first**, end, only, no, all, given, person, certain, call, same, short, long, **1**, **one**, value

Table 1: Top tokens at label positions and last position.

Label positions' value-output vectors contain concepts about the labels, and their key vectors contain the corresponding demonstrations. For example, the label at position 2 is "bar" and the value-output vector contains "BAR, Bars, Bar". Its key vector's top tokens are related to the corresponding demonstration "love". The last position have concepts about the input text "one". Hence, we hypothesize that value-output matrices extract the label information and query-key matrices compute the similarity between the last position (encodes the input text) and each label position (encodes demonstration). We also note interpretable results in sentence classification cases, detailed in Appendix A.

3.2 Identifying Important Heads for ICL

Datasets and models. We conduct the experiments on five sentence classification datasets, including financial phrasebank (Financ) (Malo et al., 2014), AG's news topic classification (AGnews) (Zhang et al., 2015), Amazon reviews (Amazon) (McAuley and Leskovec, 2013), Hate Speech Detection (ETHOS) (Mollas et al., 2020), and Stanford Sentiment Treebank binary (SST2) (Socher et al., 2013). We conduct experiments on Llama-7B (Touvron et al., 2023) with 32 layers (32 heads per layer), and GPT-J (Wang and Komatsuzaki, 2021) with 28 layers (16 heads per layer).

	Financ	AGnews	Amazon	ETHOS	SST2
foo	90.6	96.6	84.2	69.0	89.2
bar	99.8	100.0	85.6	73.2	88.8
foo	97.6	99.6	65.2	54.2	90.4
bar	98.6	83.2	98.8	92.8	97.2

Table 2: ICL accuracy (%) with correct label "foo"/"bar" in Llama (first block) and GPT-J (second block).

Inspired by Pan et al. (2023) and Wei et al. (2023) that task learning is the emergent ability of large language models (LLMs), we replace the labels with semantic-unrelated labels "foo" and "bar" to study the mechanism of ICL task learning ability. In each dataset, we randomly sample two sentences with each label, and propose the ICL sentence: **"S0 : bar S1 : bar S2 : foo S3 : foo S4 :"** with correct label "foo" and **"S0 : foo S1 : foo S2 : bar S3 : bar S4 :"** with correct label "bar", where S0 and S1 have the same label, and S2, S3, S4 have the other label. We randomly sample 1,000 sentences in each dataset. The accuracy when correct labels are "foo" and "bar" are shown in Table 2, which indicate that the ICL ability exists in most datasets.

Methods. We apply two methods to identify the important heads for ICL. Firstly, we use causal tracing methods (Pearl, 2001; Vig et al., 2020) and intervene each head in deep layers by setting the head's parameters to zero, and re-calculate the decrease in each dataset. Secondly, following Yu and Ananiadou (2024), we compute the log probability increase S_l^h of each head to find which heads directly contribute to the final predictions:

$$S_l^h = log(p(b|o_l^h + Lin_l)) - log(p(b|Lin_l)) \quad (2)$$

where b is the predicted label ("foo"/"bar"), Lin_l is lth layer's input, and o_l^h is the head output vector on layer l, head h. The probability is calculated by multiplying the vector with the unembedding matrix E_u (Eq.1). If the score is large, the head is useful for increasing the probability of label b. We identify the heads rank top10 in both methods, and there are 6 important "fooheads" affecting "foo" and 6 important "barheads" affecting "bar" in both model. The average accuracy change when intervening the fooheads and barheads is shown in Table 3. When intervening the fooheads, datasets with correct label "foo" show a significant decrease in accuracy, while those with correct label "bar" experience a substantial increase in accuracy. When masking in the barheads, datasets with correct label "bar" show a significant decrease in accuracy, while those with correct label "foo" experience a substantial increase in accuracy. Therefore, our identified fooheads and barheads are important for predicting "foo" and "bar", respectively. We name these heads "in-context heads".

	correct label : foo		correct label : bar	
	fooheads	barheads	fooheads	barheads
Llama	86.0/0.01	86.0/99.3	89.0/99.2	89.0/35.4
GPT-J	81.4/10.6	81.4/98.9	94.1/99.9	94.1/51.4

Table 3: Origin/intervened accuracy (%) when intervening fooheads and barheads in Llama and GPT-J.

3.3 Analyzing Value-Output Vectors and Attention Scores in In-context Heads

Head output o in Eq.1 is computed by the weighted sum of value-output vectors vo on all positions p:

$$o = \sum_{p=0}^{T-1} \alpha^p \cdot vo^p \quad (3)$$

where T is the length of the input text. α is the attention score computed by the softmax function on the inner product of last position's query vector and each position's key vector. vo is computed by the linear transform of value-output matrices on each position's layer input. To explore the importance of label positions in each in-context head, we investigate sentences with correct label "foo", and compute the logit minus score M at "foo" and "bar" positions' weighted value-output vectors:

$$M = log(p(foo|\alpha^p \cdot vo^p)) - log(p(bar|\alpha^p \cdot vo^p)) \quad (4)$$

If M is larger than zero, the vectors are important for enhancing "foo" probability. On the contrary, they are important for enhancing "bar" probability. The average logit minus scores at "foo" positions (fp) and "bar" positions (bp) in fooheads (fh) and barheads (bh) are shown in Table 4. In both models, foo positions contain much information about "foo" in fooheads, and bar positions contain much information about "bar" in barheads. Furthermore, the proportion between label positions' logit minus scores and the in-context heads' logit minus scores is 99.1%. Therefore, the reason fooheads/barheads affect probabilities of "foo"/"bar" is due to the information saved at "foo"/"bar" positions' weighted value-output vectors $\alpha \cdot vo$.

To explore the roles of query-key matrices and value-output matrices, we compute the attention scores and the value-output vectors' logit minus

	Financ	AGnew	Amaz	ETHOS	SST2
fh-fp	0.29	0.32	0.30	0.30	0.32
fh-bp	-0.02	-0.05	-0.04	-0.04	-0.04
bh-fp	-0.05	-0.03	-0.03	-0.02	-0.04
bh-bp	-0.11	-0.08	-0.14	-0.14	-0.12
fh-fp	0.26	0.23	0.26	0.28	0.31
fh-bp	0.00	-0.01	0.00	-0.01	-0.01
bh-fp	-0.07	-0.05	-0.06	-0.06	-0.06
bh-bp	-0.16	-0.17	-0.20	-0.23	-0.18

Table 4: Logit minus of weighted value-output vectors at "foo"/"bar" positions (fp, bp) in fooheads/barheads (fh, bh) in Llama (first block) and GPT-J (second block).

scores (removing α^p in Eq.4). The average scores across all datasets are shown in Table 5.

	fooheads		barheads	
	foopos	barpos	foopos	barpos
attn	0.742	0.047	0.369	0.195
minus	0.613	-0.574	-0.075	-0.658
attn	0.540	0.037	0.219	0.203
minus	0.958	0.099	-0.253	-1.656

Table 5: Attention score and logit minus at "foo"/"bar" positions in fooheads/barheads in Llama (first block) and GPT-J (second block), averaged on all datasets.

Both query-key matrices and value-output matrices can affect the probabilities. In Llama fooheads, the query-key matrices play large roles for predicting "foo". The value-output matrices can extract both "foo->foo" and "bar->bar", since the absolute values of logit minus scores at "foo" and "bar" positions are similar. In GPT-J fooheads, both query-key matrices and value-output matrices play large roles for enhancing "foo". In Llama barheads and GPT-J barheads, value-output matrices play larger role than query-key matrices for predicting "bar".

To explore how the predictions change from "foo" to "bar", we compare the sentences **"S0 : bar S1 : bar S2 : foo S3 : foo S4 :"** and **"S0 : foo S1 : foo S2 : bar S3 : bar S4 :"** in each dataset. We compute the change of absolute value on weighted value-output vectors' logit minus scores (minus-w), value-output vectors' logit minus scores (minus), and attention scores, shown in Table 6.

The prediction shift is caused by the change of weighted value-output vectors' logit minus scores. When changing the labels, fooheads' foo positions contain less information about "foo", and barheads' bar positions contain more information about "bar". The "foo" decrease at fooheads' "foo" positions and the "bar" increase at barheads' "bar" positions cause the probability change from "foo" to "bar".

	fooheads		barheads	
	foopos	barpos	foopos	barpos
minus-w	-12.1%	+54.4%	-47.9%	+47.2%
minus	+26.5%	-21.4%	+41.8%	-10.5%
attn	-21.8%	+124.8%	-44.4%	+91.4%
minus-w	-40.4%	+408.5%	-51.7%	+55.1%
minus	+13.9%	+32.0%	+44.2%	-17.6%
attn	-43.0%	+237.8%	-46.1%	+86.0%

Table 6: Change of attention score and logit minus at "foo"/"bar" positions in fooheads/barheads in Llama (first block) and GPT-J (second block) on all datasets.

The attention scores change significantly when the predictions shift from "foo" to "bar". Attention scores at fooheads' "foo" positions decrease substantially, while those at barheads' "bar" positions increase markedly. Comparatively, the change direction of the value-output vectors' logit minus scores does not show a relevant trend with the logit minus scores of the weighted value-output vectors. Therefore, we hypothesize that the change of attention scores within in-context heads is the primary cause for the prediction shift from "foo" to "bar".

3.4 Proposed Hypothesis and Discussion

For better understanding, we list the evidence of existing studies and previous sections: a) Wang et al. (2023) demonstrate that the label positions ("foo", "bar") extract corresponding demonstrations' features in shallow layers. b) In Section 3.2, we find that in deep layers there are a few fooheads important for predicting "foo" and barheads for "bar". c) Table 4 proves that the "foo" positions in fooheads and the "bar" positions in barheads contain much information for predicting "foo" and "bar", respectively. d) The experiments in Table 5 demonstrate that both query-key matrices and value-output matrices can affect the information storage. e) Table 6's results prove that the change of attention scores within fooheads and barheads is the primary cause for the prediction shift from "foo" to "bar" when reversing the demonstrations' labels.

Based on these findings, we conclude our hypothesis: In shallow layers, the label positions extract features from the corresponding demonstrations (hypothesized from evidence a), while the last position encodes information of the input text and previous demonstrations/labels (X% input text + Y% near demonstrations + Z% far demonstrations). In deep layers' in-context heads, the value-output matrices extract the label features into value-output vectors (hypothesized from evidence b and c). For example, fooheads extract "foo->foo" and barheads learn "bar->bar". The query-key matrices compute the similarity between the last position's features and each label position's features. When the labels change from "foo" to "bar", the change of last position features causes the similarity scores change and the prediction shift (hypothesized from evidence d and e). For instance, the fooheads' similarity scores at foo positions change from SIM((X+Y)%foo, foo) to SIM(Z%foo, foo), and the barheads' similarity scores at bar positions change from SIM(Z%bar, bar) into ((X+Y)%bar, bar). Hence, the foo positions' attention scores decrease in fooheads and the bar positions' attention scores increase in barheads, causing the probability change from "foo" to "bar".

If considering all the in-context heads together, the overall value-output matrices can learn both "foo->foo" and "bar->bar". Under our hypothesis, the query and key matrices can be regarded as two towers computing the semantic similarity between the last position's features and each label position's demonstration features. If the similarity score is large, more corresponding label information is incorporated, enhancing the probability of that label. There are four modules related to the ICL ability.

a) Information extraction ability of shallow layers. Shallow layers can be regarded as feature extraction modules. The ability of extracting corresponding demonstrations and the input text decides the quality of features.

b) Value projection ability of in-context heads' value-output matrices. If the value projection ability is good enough, the in-context heads should project "foo" and "bar" together and fairly.

c) Metric learning ability of in-context heads' query and key matrices. The query and key matrices might be the most important module, because they should learn computing different metrics using the same matrices. If different ICL tasks share the same in-context heads, the query and key matrices should learn these metrics jointly.

d) Numbers and parameters of in-context heads. If we regard one in-context head as a two-tower model for metric learning, the parameters of the head are directly related to the learning ability. At the same time, different in-context heads can be regarded as voting or ensemble models, so the head number also controls the learning ability.

4 Understanding Majority Label Bias and Recency Bias in ICL

There are several phenomena of ICL that haven't been explained. Zhao et al. (2021) demonstrate that models tend to predict majority labels and the labels near the input text. Lu et al. (2021) also find that changing the demonstration order can affect predictions a lot. Based on our hypothesis, we explore why ICL has majority label bias (in Section 4.1) and recency bias (in Section 4.2).

4.1 Understanding Majority Label Bias

According to our hypothesis, it is reasonable that the model tends to predict majority labels, because the label information flow is controlled by the similarity between last position and each label position. When a label has high frequency, the sum of similarity scores will be larger, thus the probability of this label is larger in final prediction. We design an imbalanced dataset to verify this. For each sentence with correct label "foo", we remove the last demonstration and label. For example, **"S0 : bar S1 : bar S2 : foo S3 : foo S4 :"** is changed into **"S0 : bar S1 : bar S2 : foo S4 :"**. We compute the sum of attention weights on "foo" positions in fooheads and "bar" positions in barheads on the imbalanced datasets and the original datasets, averaged on all five datasets. The changing of attention scores at "foo" positions and "bar" positions in both models are shown in Figure 2.

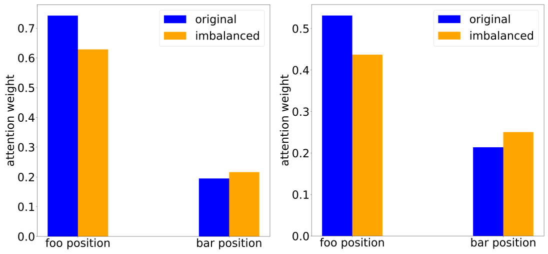

Figure 2: Attention scores on foo positions in fooheads and bar positions in barheads, on original dataset and imbalanced dataset in Llama (left) and GPT-J (right).

In both models, the sum of attention weights on "foo" positions decrease on the imbalanced dataset. On the contrary, the attention weights on "bar" positions increase. The results meet our analysis. The attention weights are computed by a softmax function, so when a "foo" demonstration and its label are removed, the sum of attention weights on "foo" positions will decrease, and that on "bar" positions will increase.

4.2 Understanding Recency Bias

The ICL performance is extremely sensitive to the demonstration order. We hypothesize that the recency bias is caused by the influence of positional embeddings on the attention score computation in both shallow layers and deep layers. The attention score is calculated by applying a softmax function to the product of the last position's query vector and each label position's key vector. These query and key vectors are derived from the layer input, which is a combination of the positional embedding, the word embedding, and the output vectors from previous attention layers and feed-forward network (FFN) layers. Therefore, a "position term" consistently influences the attention scores.

The feature extraction of last position is related to the attention scores in shallow layers' heads. Due to the influence of positional embedding, the model tends to extract varying amounts of features at different positions. Let us consider the case **"S0 : bar S1 : bar S2 : foo S3 : foo S4 :"**. The last position contains X% S4 + Y% (S2+S3) and Z% (S0+S1), simplified into (X+Y)% foo + Z% bar. If the demonstration order is changed into **"S2 : foo S3 : foo S0 : bar S1 : bar S4 :"**, the last position will contain X% S4 + Z% (S0+S1) + Y% (S2+S3), simplified into (X+Z)% foo + Y% bar. Hence, the final prediction probability will be different between these two sentences if Y and Z are different. If Y is larger than Z, the last position will contain less "foo". Similarly, the influence of positional embeddings also exists in deep layers' heads, which tends to enlarge the attention scores on later positions in these heads.

We design a reverse dataset to evaluate the difference among different positions. For each sentence **S0 : bar S1 : bar S2 : foo S3 : foo S4 :**, we transfer it into a reverse sentence **S2 : foo S3 : foo S0 : bar S1 : bar S4 :**. We compute the average attention score change at "foo" positions in fooheads and "bar" positions in barheads, between the original and the reverse dataset, shown in Figure 3. Moreover, we remove the impact of positional embedding in each in-context head and re-compute the attention scores (original modify and reverse modify in Figure 3).

Compared with the original dataset, "foo" positions' attention weights decrease and "bar" positions' attention weights increase in the reverse dataset in both models. This result aligns with the observations in previous studies (Zhao et al., 2021) that the probability is affected much when reversing the demonstration order. When removing the impact of positional embedding in each head, the near positions' attention scores decrease and the far positions' scores increase. Hence, our hypothesis is verified: the positional term in each head enlarges the attention scores on later positions. After removing the positional term in in-context heads, the attention score is still different between the original dataset and the reverse dataset. This difference is caused by the difference in shallow layers' feature extraction stage.

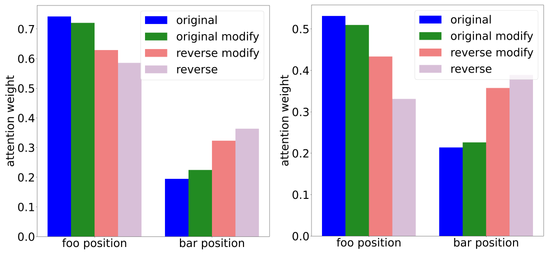

Figure 3: Attention scores on foo positions in fooheads and bar positions in barheads, on original dataset and reverse dataset in Llama (left) and GPT-J (right).

To provide a clearer perspective, we illustrate the attention score change on "foo" positions in each foohead and "bar" positions in each barhead. The change of imbalanced dataset and reverse dataset in Llama and GPT-J is shown in Figure 5 and 6, where the first 6 columns are "foo" positions' attention scores in fooheads and the last 6 columns are "bar" positions' scores in barheads. Compared with the original dataset, the attention scores decrease on "foo" positions and increase on "bar" positions in imbalanced dataset and reverse dataset.

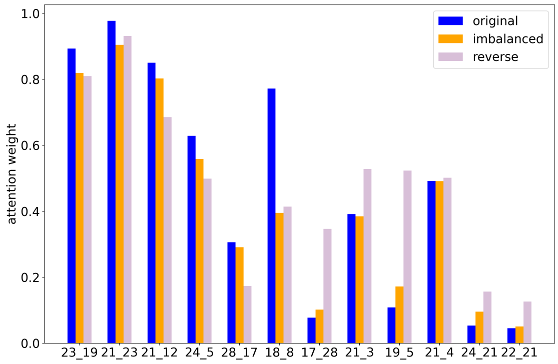

Figure 4: Attention scores on "foo"/"bar" positions in original, imbalanced, and recency datasets in Llama.

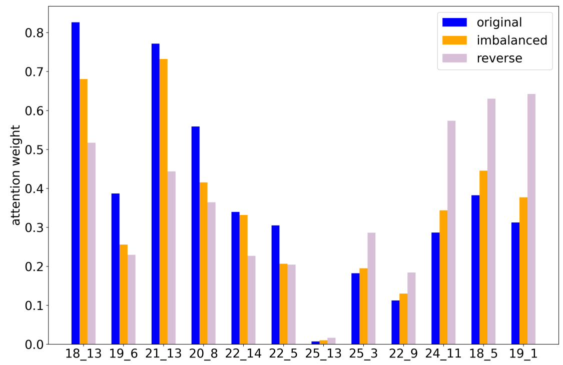

Figure 5: Attention scores on "foo"/"bar" positions in original, imbalanced, and recency datasets in GPT-J.

5 Reducing Majority Label Bias and Recency Bias

In this section, we propose a method for reducing majority label bias in Section 5.1, and propose a method for reducing recency bias in Section 5.2.

5.1 Reducing Majority Label Bias by Enlarging Imbalanced Label Attention

According to our analysis in Section 4.1, the majority label bias can be attributed to the lack of attention weights on imbalanced label positions. So we propose a method to reduce the majority label bias by enlarging the imbalanced label positions' attention scores. Specifically, we multiply an amplified score a on the imbalanced label positions' weighted value-output vectors ($a\alpha^p \cdot vo^p$ in Eq.3) and add this vector into the final embedding. a is the product of a constant hyperparameter a_c and a varying score a_v, where a_v is the ratio of the larger demonstration number to the smaller demonstration number.

We first make a balanced dataset by randomly sampling 2-4 demonstrations in each label, and randomly set the demonstration order. The correct labels of the balanced sentences are "foo". Then we get a "lackfoo" sentence by randomly removing a "foo" demonstration, and a "lackbar" sentence by randomly removing a "bar" demonstration. Except the results in Financ GPT-J, the accuracy of "lackfoo" dataset is smaller than the balanced dataset due to the lack of "foo" demonstrations, and "lackbar" accuracy is larger than the balanced dataset.

Compared to the balanced dataset, we calculate the sum of accuracy change on "lackfoo" and "lackbar" datasets before and after applying our method with amplified constant score a_c 0.03. The accuracy change is shown in Table 7. On average, the accuracy change reduces 29.1% in Llama and 14.9% in GPT-J. The results indicate that our method can reduce the accuracy change caused by the influence of imbalanced demonstrations/labels.

	Financ	AGnew	Amaz	ETHOS	SST2
before	0.10	0.09	0.23	0.10	0.19
after	0.07	0.05	0.17	0.07	0.15
before	0.04	0.03	0.05	0.05	0.08
after	0.06	0.02	0.02	0.03	0.06

Table 7: Accuracy change before/after applying our method in Llama (first block) and GPT-J (second block).

5.2 Reducing Recency Bias by Removing Positional Embedding Affect

As discussed in Section 4.2, we find the recency bias is due to the effect of positional embedding on the calculation of attention scores. Hence, in order to reduce the recency bias, we reduce the position term in in-context heads, and re-calculate the output vectors in all in-context heads. This method is similar with adding a shortcut adapter from each in-context head to the final embedding.

	Financ	AGnew	Amaz	ETHOS	SST2
acc-be	0.37	0.42	0.26	0.22	0.30
acc-af	0.31	0.39	0.15	0.16	0.18
attn-be	0.06	0.08	0.06	0.05	0.06
attn-af	0.03	0.06	0.04	0.03	0.03
acc-be	0.39	0.27	0.45	0.41	0.40
acc-af	0.36	0.16	0.42	0.40	0.35
attn-be	0.07	0.05	0.07	0.06	0.08
attn-af	0.04	0.03	0.05	0.04	0.05

Table 8: Standard deviation of accuracy and attention scores before/after applying our method in Llama (first block) and GPT-J (second block).

We apply this method to the original dataset and three recency datasets with different demonstration orders, detailed in Appendix B. We calculate the standard deviation in accuracy and in-context heads' attention scores before (acc-be, attn-be) and after (acc-af, attn-af) applying our method. The results are shown in Table 8. On average, the accuracy standard deviation reduces 23.4% in Llama and 10.6% in GPT-J, and the attention score standard deviation reduces 40.1% in Llama and 37.7% in GPT-J. Therefore, removing the positional term in in-context heads is helpful for reducing the recency bias. It is also important to reduce the recency bias during feature extraction in shallow layers, and we leave this exploration in future work.

6 Conclusion

We identify the important heads for ICL and analyze the value-output vectors and attention scores in these heads. We propose a hypothesis for the mechanism of ICL. In shallow layers, the demonstrations and input text is captured by the label positions and the last position. In in-context heads, the value-output matrices project the label features into value-output vectors. The query and key matrices can be regarded as two towers learning the similarity between the last position's features and each label position's features. If the similarity score is high, the corresponding label's probability is enlarged. Based on this hypothesis, we interpret why ICL has majority label bias and recency bias. Furthermore, we propose two methods to reduce these biases by 22% and 17%. Overall, our study provides a new method and a reasonable hypothesis for understanding the mechanism of in-context learning.

7 Limitation

In this paper, we focus on understanding the mechanism in in-context heads in deep layers. It is also important to study how shallow layers transfer features into label positions and the last position. Our hypothesis explains the ICL mechanism for classification tasks. More studies should be done on other ICL tasks, such as chain-of-thought reasoning (Wei et al., 2022b).

Another limitation of our work comes from the attribution method for identifying important heads. Gradient-based methods and causal tracing methods, which calculate a module's impact on the final prediction, are commonly employed for importance attribution. Additionally, many studies utilize saliency score-based methods. In this paper, we apply both causal tracing and saliency score-based methods to identify important heads, and we believe the results in Table 3 support our findings. However, it is important to note that there is no unified method for attributing important modules, and further exploration is needed to design better attribution methods.

8 Acknowledgements

This work is supported by the project JPNP20006 from New Energy and Industrial Technology Development Organization (NEDO). This work is supported by the computational shared facility and the studentship from the Department of Computer Science at the University of Manchester.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661.*

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559.*

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535.*

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234.*

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680.*

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037.*

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *arXiv preprint arXiv:2305.00586.*

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction. *arXiv preprint arXiv:2303.05063*.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 267–280.

Jia Li, Yunfei Zhao, Yongmin Li, Ge Li, and Zhi Jin. 2023a. Towards enhancing in-context learning for code generation. *arXiv preprint arXiv:2303.17780*.

Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. 2023b. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Nostalgebraist. 2020. Interpreting gpt: the logit lens.

Chris Olah. 2022. Mechanistic interpretability, variables, and the importance of interpretable bases. In *Transformer Circuits Thread*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. *What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning*. Ph.D. thesis, Princeton University.

Judea Pearl. 2001. Direct and indirect effects. *Probabilistic and Causal Inference: The Works of Judea Pearl*, page 373.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Suzanna Sia and Kevin Duh. 2023. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. *arXiv preprint arXiv:2305.03573*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. *arXiv preprint arXiv:2312.12141*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Case Study on Sentence Classification

We analyze a sentence classification case sampled in AGNews dataset. The top tokens in head 23-13 in GPT2 large are shown in Table 9. With the prediction "foo", the case is:

Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street's dwindling band of ultra-cynics, are seeing green again. : **bar** Stoking the Steamroller No other recording artist can channel American middle-class tastes quite like Chip Davis and his best-selling band. : **bar** Liverpool completes signings of Alonso, Garcia LIVERPOOL, England (AP) – Spanish pair Xabi Alonso from Real Sociedad and Luis Garcia from Barcelona signed five-year contracts with Liverpool on Friday. : **foo** U.S. Doping Watchdog to Question BALCO's Conte - IAAF HELSINKI (Reuters) - U.S . anti-doping officials plan to question Victor Conte after the BALCO head claimed he saw sprinter Marion Jones taking banned drugs, world athletics body the IAAF said Saturday. : **foo** Liverpool Progresses to Champions League; Monaco, Inter Advance Four-time champion Liverpool progressed to soccer Champions League 2-1 on aggregate, overcoming a 1-0 home defeat to AK Graz in the second leg of qualifying. **:**

position	**top words in vocabulary space**
bar-value	**BAR**, **bars**, **Bars**, bart, **Bar**, bartender, **bar**, Barber
bar-value	bartender, **Bars**, bart, **bars**, **Bar**, Barber, **bar**, **BAR**
foo-value	**foo**, McKenzie, **Foo**, Barney, Walters, Jenner, Murphy, lobster, Handler
foo-value	Walters, **foo**, Barney, McKenzie, Harrington, Murphy, Barber, Barron, Jenner
bar-key	Bloomberg, Investor, billionaires, CNBC, bankers, Companies, JPMorgan, obal, economists, bullish, Barron, HSBC, Friedman, Consumer, business, sellers
bar-key	Buy, Conn, Ok, Previous, Daily, NY, Yes, Anon, US, Ibid, Profit, Staff, Journal, Vanguard, Tribune, Well
foo-key	Buy, iverpool, Ibid, YORK, UNITED, Oliv, Charl, Location, Spanish, Miami, US, Liverpool, Pool, London, Greenwich, United
foo-key	NYT, WATCH, Latest, Exclusive, Previous, UNC, US, Watch, Possible, Ibid, Statement, Reaction, UK, Reuters, United, Smoke
last-query	ruary, Pipe, lihood, swick, Flavoring, iverpool, paddle, paraph, Lake, Repe, tong, bole, etheless, Lakes

Table 9: Top words of labels and last token in GPT2 large layer 23, head 13 on a sentence classification case.

In this case, the false demonstrations with label "bar" are sampled from the "Business" class. The true demonstrations with label "foo" and the input text are sampled from the "Sports" class. On label positions' value-output vectors, "bar" and "foo" have top rankings. As for the key vectors at label positions, the labels correspond to business demonstrations extract the concepts about business, such as "investor" and "profit". The top tokens of true labels are related to places such as "Liverpool" and "Spanish", which exist in the corresponding demonstrations. These observations indicate that the value-output matrices extract label features, and the key matrix extract corresponding demonstration features. Analyzing the last position's query vector, we also observe concepts related to "Liverpool".

B Recency Datasets for Evaluation

The three recency sentences transformed from the original sentence is shown in Table 10.

	sentence
origin	**S0 : bar S1 : bar S2 : foo S3 : foo S4 :**
reorder-1	**S2 : foo S0 : bar S1 : bar S3 : foo S4 :**
reorder-2	**S0 : bar S2 : foo S3 : foo S1 : bar S4 :**
reverse	**S2 : foo S3 : foo S0 : bar S1 : bar S4 :**

Table 10: Sentences transferred from origin sentence.