Detecting Generalization Barriers for Understanding Neural Machine Translation

Anonymous EMNLP submission

Abstract

In machine translation evaluation, the traditional wisdom measures model's generalization ability in an average sense, for example by using corpus BLEU. However, the statistics of corpus BLEU cannot provide comprehensive understanding and fine-grained analysis on model's generalization ability. As a remedy, this paper attempts to understand NMT at word level, by detecting generalization barrier words within an unseen input sentence that *cause* the degradation of generalization. It proposes a principled definition of generalization barrier words as well as a modified version which is tractable in computation. Based on the modified one, three simple methods are proposed for barrier detection by search-aware risk estimation through counterfactual generation. Extensive analyses are conducted on those detected generalization barrier words on both Zh⇔En NIST benchmarks. Potential usage of barrier words is also discussed.

1 Introduction

The performance of neural machine translation (NMT) models has been boosted significantly through novel architectural attempts (Gehring et al., 2017; Vaswani et al., 2017), carefully-designed learning strategies (Ott et al., 2018), and semi-supervised techniques that smartly increase the size of training corpous (Edunov et al., 2018; Ng et al., 2019). However, all these improvements are measured in an *average* sense on a held-out dataset by using corpus BLEU (Papineni et al., 2002) and one potential limitation may stand out for understanding NMT. The average case analysis only covers the mean data population and does not provide much *fine-grained* information on questions like why an unseen input hinders model's generalization and what properties such an input has, which are important to understand NMT and receiving great attention in the community of trustworthy deep learning (Amodei et al., 2016; Jia et al., 2019).

One possible solution to mitigate the above limitation is to analyze the property of the unseen input sentence *as a whole* as the so-called *instance*-level analysis. This is similar to recent renaissance of out-of-distribution detection in the task of image classification (Chandola et al., 2009; Hendrycks and Gimpel, 2017; Liang et al., 2017). Nevertheless, for the task of machine translation, since an input sentence consists of many words, it is not reasonable to regard the whole sentence as an anomaly since we find that the overall generalization of the model is mostly affected by a few words and modifying them can improve translation quality largely. This phenomenon is shown in Figure 1, where by changing ***quēxiàn*** to some other words, the input sentence can be translated much better, reaching better sentence-level BLEU in the orange band in Figure 1. Therefore, it would be more appropriate to automatically detect those generalization barrier words for understanding NMT at *word* level, e.g. the words within an input which hinder the overall generalization of that sentence.

To this end, we firstly give a principled definition of generalization barrier in a counterfactual (Pearl and Mackenzie, 2018) way for understanding NMT at *word* level. Since the principled definition requires human evaluation, we instead provide a modified definition based on novel statistics, which employ automatic evaluation to detect generalization barrier words. As it is costly to exactly compute the statistics, we propose three estimators to approximately calculate the value. Based on the estimated value, we conduct experiments on two benchmarks to detect potential barriers in each unseen input sentence. In addition, we carry out systematic analyses on the detected barriers from different perspectives. We find that generalization barrier words are pervasive among different linguistic categories (Part-of-

Speech) and very different from previously known troublesome source words (Zhao et al., 2018, 2019). By aggregating local barrier statistics, we find that barrier words are very context-sensitive, so they might be inevitable from current training paradigm. Moreover, the notion of barrier words motivates us to obtain more diversified hypothesis candidates via input editing. This might be a better choice for re-ranking (Yee et al., 2019) than the top-k outputs under one steady input via beam search.

2 Related Literature

Troublesome words detection To our knowledge, back to the old SMT era, Mohit and Hwa (2007) is the most related work which invents the notion of 'hard-to-translate phrase' at source side, and uses removal to determine its effect on model generalization on other phrases' translation, which is very similar to our usage of counterfactual generation by editing the source words. Recently, Zhao et al. (2018, 2019) are the first to detect *trouble makers* at source side globally for NMT. In Zhao et al. (2018), the troublesome source words are detected through an exception rate defined as the number of troublesome alignments (x_i, y_j) dividing the number of x_i, where the troublesome alignments are obtained through an extrinsic statistical aligner instead of the trained NMT model. In Zhao et al. (2019), the troublesome source words are constrained to words with high translation entropy which tend to be under-translated by the model. Both of their trouble detection heuristics are: 1) context-unware, globally applied on every source words without considering the context of the words, and 2) model-unaware, dependent on extrinsic statistical assumptions. In our work, we are trying to detect both context-aware and model-specific generalization barriers for every unseen source input.

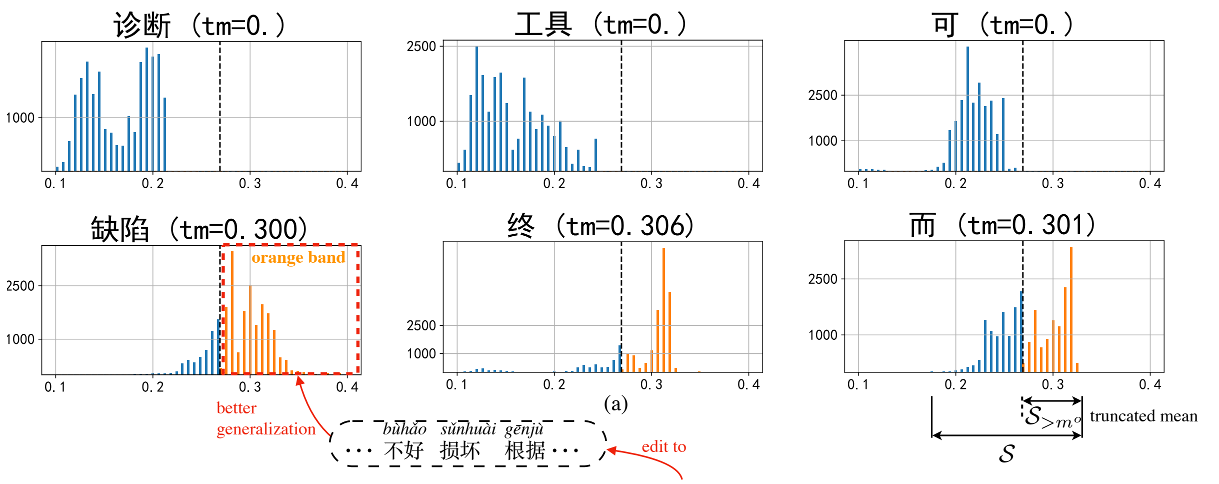
Out-of-Distribution (OOD) detection OOD detection, Novelty (Markou and Singh, 2003), Outlier (Hodge and Austin, 2004) or Anomaly Detection (Chandola et al., 2009) care about how likely the unseen input *as a whole* is to be sample different from the training distribution. This problem is recently revived on the task of image classification (Hendrycks and Gimpel, 2017; Liang et al., 2017; Choi et al., 2018). Although recently, Ren et al. (2019) starts to consider OOD detection on sequential data, i.e. gene fragments, they still regard the input feature as a holistic vehicle to cause the mismatch in underlying generative distribution. Our work is motivated from this OOD detection literature in the spirit of detecting the inputs that the model cannot generalize well upon. Beyond that, due to the structural property of the translation task, we also carry out a more fine-grained detection of *causes* that could be a part of the input feature, which can potentially consist of several high risky words. Notably, researchers from OOD detection recently start to focus on structure of the input and design benchmarks for such detection task for image anomaly segmentation which focuses on small patches in the image (Hendrycks et al., 2019).

Error analysis and interpretability Recently, Wu et al. (2019) propose to conduct error analysis with three principles by heart: scalable, reproducible and counterfactual for natural language processing tasks. These principles also guide the computational consideration of our detection method. For NMT, recently, Lei et al. (2019) are the first to focus on accurately detecting wrong and missing translation of certain source words. Different from their work which detects the unsatisfactorily translated source words themselves, our work focuses on detecting the *cause* of them, and serves as complementary to recent interpretability analysis of importance words (He et al., 2019).

3 Generalization Barriers

Mainstream NMT is formulated as a sequence-to-sequence structured prediction problem (Sutskever et al., 2014). Like all other structured prediction problems with a scoring function and a decoding algorithm (Daumé III, 2006), for NMT, $P(\mathrm{y}|\mathrm{x};\theta)$ acts as the scoring function and beam search is used as the (approximate) decoding algorithm. Since beam search is a deterministic algorithm with a preset beam size, the prediction $\hat{\mathrm{y}}$ is *solely* determined by the input x, denoted as a map $\hat{\mathrm{y}} = \mathcal{M}_{\hat{\theta}}(\mathrm{x})$. Under this setting, we are actually interested in the following **causal question**: *how the input* x *causes the model's failure on the prediction?*

The input of NMT model $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_m)$ is a sequence with subsequences composed to form its whole semantics. The cause of the model's generalization degradation should be attributed to some of the subsequences or their ways of composition. For example, in Figure 1, by changing the subsequence *quēxiàn* to another word (for examples, *bùhǎo*, *sǔnhuài* or *geñjùn*) can make the not-well translated subsequences (marked as <u>underlined</u> subsequences in the original hypo) to be well-translated. There-



Source	基因 科学家 的 目标 是 , 提供 诊断 (zhěnduàn) 工具 以 发现 致病 的 ~~缺陷~~ (quēxiàn) 基因 , 终 而 提供 可 阻止 这些 基因 产生 障碍 的 疗法 。
Reference	the goal of geneticists is to provide diagnostic tools to identify defective genes that cause diseases so as to arrive eventually at treatments that can *prevent those genes from malfunctioning* .
Original Hypothesis	the goal of genetic scientists is to provide diagnostic tools to discover the defects of the disease and , in the end , to provide treatments that can *prevent the production of these genes* .
Much better Hypothesis	the goal of gene scientists is to provide diagnostic tools to detect genes that cause diseases , and eventually provide treatments that can *prevent these genes from producing obstacles* .

(b)

Figure 1: **(a)**. Each histogram represents $|\mathcal{V}|$ sentence-level BLEU scores via $|\mathcal{V}|$ number of editing choices on the source word at the top. The x-axis is the BLEU spectrum 0.1-0.4 divided into 50 bins; the y-axis denotes the number of edits that fall into certain BLEU bin; the vertical dotted line is the metric value of the original hypothesis. **(b)**. A list of the example source sentence, its reference, the original hypothesis and the improved hypothesis via editing the generalization barrier word *quēxiàn*. (*This figure is better viewed with color.*)

fore, one perspective to shed light on the above *how* question is to try to *detect* the set of all subsequences of x that might potentially deteriorate model's generalization, which we dub *generalization barriers* (such as *quēxiàn* in Figure 1). In the following subsections, we firstly give an abstract yet principled definition of generalization barriers. Then we relax this definition to obtain an approximate but tractable version by treating each source word independently without considering their possible combinatorial compositions. Finally we construct statistics for each source word to measure its risk of being a generalization barrier word.

3.1 A definition with human effort

The principled definition of generalization barriers is based on the intuition that the model can potentially generalize well on some edited versions of x, i.e. with word *substitutions* and *deletions* that aims at preserving the original symbolic compositional structure (e.g. word order) and semantics of x as much as possible. This intuition also matches with the causal question we have asked before, since we are actually generating counterfactuals through intervening (editing) x (Chang et al., 2018; Goyal et al., 2019). Formally, we define generalization barrier words in x as follows.

Definition 3.1. (*Generalization Barriers*) *Given an NMT model trained on $\mathcal{D}^{tr}$ with $\hat{\theta}$, a distance measure $d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, e.g. the edit distance, for an input* x*, we call the set of subsequences,* $\cup(\mathrm{x} \setminus \tilde{\mathrm{x}})$*, that satisfy the following constraints as generalization barriers of* x.

1. *The distance measure $d(\mathrm{x}, \tilde{\mathrm{x}})$ is minimized;*

2. *Human evaluation of the translation quality on $\mathcal{M}_{\hat{\theta}}(\tilde{\mathrm{x}})$ reaches a satisfactory level.*

where the operator $\setminus$ returns a subsequence of x by removing their overlapped words and $\cup$ denotes the union of subsequences. As an example in Figure 1, editing *quēxiàn* to each word among *bùhǎo*, *sǔnhuài* and *gēnjùn* leads to much better translation. Therefore, $\tilde{\mathrm{x}}$ can denote the source by editing *quēxiàn*, and the generalization barrier word of x is $\mathrm{x} \setminus \tilde{\mathrm{x}} = \{quēxiàn\}$, including a single word.

Although this definition requires human efforts, we think it is principled and can be of independent interest for human study.

3.2 Approximating the definition with exhaustive counterfactuals

To scale up the above definition, we further make assumptions to modify it: a) the minimization of d is purposefully set to $d = 1$, which restricts the search space tremendously by only editing one word for investigating its possibility of being a barrier; b) the human evaluation is replaced by automatic evaluation with a metric such as smoothed sentence-level BLEU (Lin and Och, 2004), since $d = 1$ roughly leads to an unchanged reference y.

According to the modification, we now investigate each source word x_i independently by counterfactual generation as well. Instead of finding one single counterfactual $\tilde{\mathrm{x}}$ which might be unsuitable for human to perceive as a natural sentence, inspired by Burns et al. (2019) and Chang et al. (2018) who edit certain patch in an image with potentially *infinitely* infilling patches and compute importance score of the original patch in expectation, we also generate as many edit choices as possible so that some edits are *natural*.

Suppose $|\mathcal{V}|$ is the size of the source vocabulary, $\text{Edit}(\mathrm{x}, i)$ denotes the set of all sentences by editing word x_i. Accordingly, the size of $\text{Edit}(\mathrm{x}, i)$ is $|\mathcal{V}|$, which corresponds to one deletion and $|\mathcal{V}|$ - 1 substitutions. Then we can actually obtain $|\mathcal{V}|$ counterfactual performance measures:

$$\mathcal{S} = \{\text{BLEU}(\mathcal{M}_{\hat{\theta}}(\tilde{\mathrm{x}}), \mathrm{y}) | \tilde{\mathrm{x}} \in \text{Edit}(\mathrm{x}, i)\}, \quad (1)$$

based on which we can draw a *histogram* with binned metric values, with vertical axis denoting the *number* of edits that can lead to certain performance. Figure 1 is a showcase for a given input sentence, we conduct $|\mathcal{V}|$ real decoding for each of the 28 words and plot the corresponding histograms of six words, with the first row represents words that always degrade the performance, and second row words mostly improve performance through editing. We regard each histogram as a distribution of the counterfactual generalization performances.

As we can identify in Figure 1, the right-hand side orange band of a histogram (if exists) shows the counterfactuals with better generalization, and if that part *dominates* the distribution, we can conclude that the word being edited has a *high risk* of *causing* the degradation of generalization on x. In practice, we use the empirical truncated mean at position i to represent the word x_i's risk of being a generalization barrier word as follows:

$$tm(\mathrm{x}_i) = \frac{1}{|\mathcal{S}_{>m^o}|} \sum_{v \in \mathcal{S}_{>m^o}} v, \quad (2)$$

where $\mathcal{S}_{>m^o} = \{v | v > m^o, v \in \mathcal{S}\}$ and $m^o = \text{BLEU}(\mathcal{M}_{\hat{\theta}}(\mathrm{x}), \mathrm{y})$. In Figure 1, the set $\mathcal{S}_{>m^o}$ corresponds to the orange band in the histogram; m^o is 0.262 in the example, corresponding to the dotted black vertical line in each histogram; the truncated mean of each word is presented above the histogram, for example, the word *quēxiàn* has a high truncated mean of 0.30 which is above 4 BLEU points than the original performance. The higher the risk, the more likely that word being a generalization barrier word. In Figure 1, the truncated mean (tm) is shown above each histogram, with the value 0 denotes that position has *no* orange band.

Definition 3.2. (*Generalization Barrier Words*) *The generalization barrier words in* x *tend to be the words with top-k% truncated mean $tm(\mathrm{x}_i)$.*

In practice it is hard to determine whether a truncated mean reaches a satisfactory level, so we use a soft one, the top-k% risky words, for deciding the potential generalization barrier words.

Algorithm 1: Evaluate the risk of x_i

Input:
A risk estimator S;
an unseen pair x, y, position i, budget B, b;
the learned NMT model $P(\mathrm{y}|\mathrm{x}; \hat{\theta})$,
the source embedding $\text{Emb} \in \mathbb{R}^{|\mathcal{V}| \times d}$;
Output:
The estimated truncated mean $tm(\mathrm{x}_i)$;

```
Initialize C^i = {};
if S = Uniform then
    Uniformly sample b elements from Edit(x, i), and add them to C^i;
else if S = Stratified then
    Uniformly sample B elements from Edit(x, i) as C^i_0;
    Compute L_θ̂(x̃) in Eq.(3) for each x̃ ∈ C^i_0;
    Use s_x̃ ∝ 1/L_θ̂(x̃) to choose the top-b elements in C^i_0, and add them to C^i;
else if S = Gradient-aware then
    Compute Eq. (4) to get Emb'(x_i);
    Use softmax(Emb · Emb'(x_i)) to sample b elements from Edit(x, i), and add them to C^i;
end if
Conduct real decoding on C^i and compute tm(x_i) supported on C^i rather than Edit(x, i).
return tm(x_i);
```

3.3 Estimating truncated mean

According to the definition in Eq.(2) and Eq.(1), one has to decode each $\tilde{\mathrm{x}} \in \text{Edit}(\mathrm{x}, i)$ and there are

$|\mathcal{V}|$ sentences in total. Unfortunately, as it takes a few seconds for each decoding, it is impractical to exactly calculate $\mathcal{S}$ as well as $\mathcal{S}_{>m^o}$ As a result, we instead propose a simple yet effective algorithm as an inexact solution. The key idea to the inexact solution is to call the decoder b times, with b as a budget. Specifically, we randomly sample b elements from $\mathrm{Edit}(\mathrm{x}, i)$ to obtain a sample set C^i. Then we calculate both $\mathcal{S}$ and $\mathcal{S}_{>m^o}$ supported on C^i. Finally we can approximately calculate $tm(\mathrm{x}_i)$ by enumerating at most b elements in $\mathcal{S}_{>m^o}$. To randomly sample b elements from $\mathrm{Edit}(\mathrm{x}, i)$, we pre-define three distributions heuristically, which lead to three different estimators as follows.

Uniform A very simple unbiased estimator of $tm(\mathrm{x}_i)$ is to uniformly b elements from $\mathrm{Edit}(\mathrm{x}, i)$, and compute the mean of those ms that are larger than m^o. However, since we do not restrict the substitutions, two potential issues might lead to large variance of uniform sampling: a) waste of budget: substitutions that lead to metric values lower than m^o could be more; b) hardness of coverage (less concentrated): wider the range of the orange band (in the histogram of Figure 1), larger the variance.

Stratified To be less stochastic to combat variance, we can first use uniform sampling for randomly picking B elements from $\mathrm{Edit}(\mathrm{x}, i)$, and then use the loss function

$$\mathcal{L}_{\hat{\theta}}(\tilde{\mathrm{x}}) := -\log P(\mathrm{y}|\tilde{\mathrm{x}}; \hat{\theta}) \qquad (3)$$

as a surrogate to choose the top-b from the B choices. The first stage respects the uniform distribution in $\mathrm{Edit}(\mathrm{x}, i)$, while the second stage is deterministic (i.e., top-b likelihood values) which can potentially lower the variance.

Gradient-aware To avoid the sampling budget hyper-parameter B at the first stage of the stratified method, we can utilize the gradient of the original loss $\mathcal{L}_{\hat{\theta}}(\mathrm{x})$ which guides the change of embeddings of x_i that can minimize the loss:

$$\mathrm{Emb}'(\mathrm{x}_i) = \mathrm{Emb}(\mathrm{x}_i) - 1.0 \cdot \nabla_{\mathrm{Emb}(\mathrm{x}_i)} \mathcal{L}_{\hat{\theta}}(\mathrm{x}). \quad (4)$$

Contrary to the method of adversarially modifying the input in Cheng et al. (2019), we conduct 1-step gradient update with learning rate 1.0 to *minimize* the original loss, and then use the normalized dot product similarity between the updated embedding and all other embeddings of the source vocabulary to bias the sampling of b elements from $\mathrm{Edit}(\mathrm{x}, i)$.

The entire algorithmic procedures of the three estimators are summarized succinctly in Algorithm 1.

	second per sentence							
budget b	5	10	25	50	100	250	500	1000
time cost	4	7	17	33	65	180	360	> 600

Table 1: The time complexity for the uniform estimator among different budgets; note that the time cost is an average measure over each sentence.

4 Experimental Conditions

Data settings We conduct experiments on Zh⇒En and En⇒Zh translation tasks using the well-known NIST benchmark. The development and test datasets of the NIST benchmark are marked by year, e.g. NIST02 (dev), NIST03 etc. For Zh⇒En, each dev/test source sentence has four references; and for En⇒Zh, we pick the first source input of the four as the source-side instance. During the truncated mean estimation stage, for the Zh⇒En translation task, we use the first reference as the ground truth in sentence-level BLEU calculation.

Model settings We consider three types of basic model architectures proposed in Luong and Manning (2015); Gehring et al. (2017); Vaswani et al. (2017) respectively, representing the advancement of architectural inductive bias in recent years. Their average performance over NIST03, 04, 05, 06, 08 are summarized in Table 7 in Appendix A.1.

5 Comparing Estimators [1]

We conduct simulation experiments among 50 unseen sentence pairs from NIST03 with whole vocabulary decoding to compute the ground truth truncated mean for each x_i with Eq. (2), and then compare the above proposed sampling methods in terms of *overlap@k%*, *variance* or *rank stability* of the estimator under different budgets $b = 5, 10, 25, 50, 100, 250, 500, 1000, 5000$. For the stratified strategy, we set B to 500 for $b < 500$ budgets, $B = 1000$ for $b = 500$, $B = 2000$ for $b = 1000$, and $B = 10000$ for 5000. To be statistically significant, for each source word x_i, we repeat the estimation procedure for $r = 25$ times.

Accuracy We use the overlap@$k\%$ metric to measure the similarity between top-$k\%$ risky words with exact and approximate risk calculation methods. As demonstrated in Figure 2, different methods lead to very overlapped performance. And with a budget larger than 100, it can lead to an average overlap@$k\%$ around 85%, based on which

[1] More detailed informations about the evaluation metrics used in this subsection are in Appendix A.2.

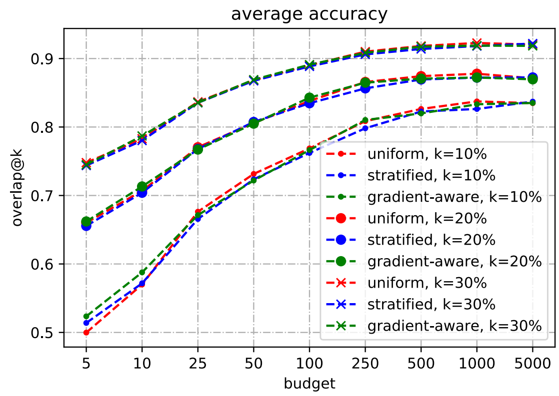


Figure 2: The overlap@k% metric values over the three proposed estimation methods on the 50 samples under different budgets (5 to 5000); k is set to 10, 20, 30(%).

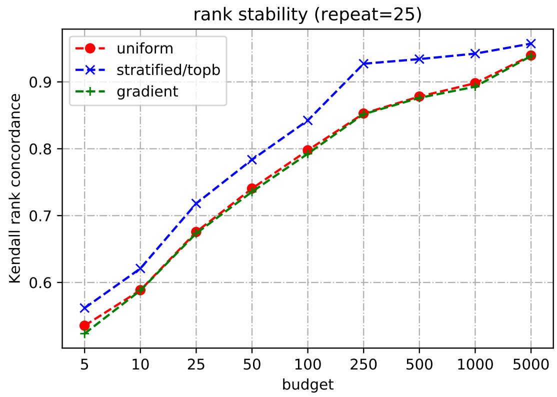


Figure 3: The rank stability of the three proposed estimation methods under different budgets. They are averaged over the 50 chosen samples and measures the variance of methods over 25 repeated experiments.

we think is enough for the subsequent analyses.

Variance The rank stability is measured through Kendall's coefficient of concordance (Mazurek, 2011) which essentially calculates the similarity among different (repeat=25) ranks of the same sample. The larger the value is, the more consistent among different runs the ranks stay, thus smaller variance. In Figure 3, the uniform and gradient-aware estimators have similar variance while the stratified estimator has lower variance, which might be benefited from its second deterministic stage.

Complexity We also summarize the time cost of each budget b in Table 1. Since most of the time complexity comes from real decoding, here we only measure the time cost of the uniform estimator. We test the process on a single M40 GPU.

As a trade-off between accuracy, variance and time complexity, we adopt the stratified strategy with budget $B = 500, b = 100$ as our approximate

POS cat.	k=10%	k=20%	k=30%	base
BPE	14.32%$^-$	15.10%$^-$	15.28%$^-$	15.33%
Noun	16.52%$^-$	16.23%$^-$	15.83%$^-$	17.63%
Prop. N.	6.56%$^-$	6.75%$^-$	6.37%$^-$	7.44%
Pron.	1.75%$^-$	1.91%$^-$	2.32%$^-$	2.35%
Verb	18.37%$^+$	18.33%$^-$	18.56%$^+$	18.36%
Adj.	2.50%$^-$	2.56%$^-$	2.60%$^-$	3.19%
Adv.	4.30%$^+$	4.27%$^+$	4.14%$^+$	4.07%
Prep.	4.70%$^+$	4.65%$^+$	4.58%$^+$	3.83%
Punc.	16.65%$^+$	14.49%$^+$	14.40%$^+$	11.44%
Q&M	3.95%$^-$	4.49%$^-$	4.59%$^-$	4.87%
C&C	1.84%$^-$	1.79%$^-$	1.99%$^-$	2.23%

(a) on NIST03 Zh⇒En direction

POS cat.	k=10%	k=20%	k=30%	base
BPE	9.80%$^-$	10.74%$^-$	11.26%$^-$	12.00%
Noun	22.17%$^-$	22.43%$^-$	21.85%$^-$	24.07%
Pron.	1.94%$^-$	2.18%$^+$	2.26%$^+$	2.15%
Verb	11.57%$^+$	11.28%$^+$	11.00%$^-$	11.26%
Adj.	6.74%$^-$	7.19%$^-$	7.26%$^-$	8.19%
Adv.	3.24%$^+$	3.07%$^+$	2.83%$^-$	2.93%
Prep.	12.94%$^+$	13.05%$^+$	13.39%$^+$	11.88%
Punc.	16.04%$^+$	13.98%$^+$	13.30%$^+$	10.41%
Det.	8.11%$^-$	8.84%$^-$	9.42%$^+$	9.05%
C&C	1.94%$^-$	2.06%$^-$	2.05%$^-$	2.20%

(b) on NIST03 En⇒Zh direction

Table 2: Distribution of the detected generalization barrier words according to POS category.

detection method in all subsequent analyses. This takes around 16 hours for $1k$ sentences with a decent detection accuracy around 85% with respect to overlap@k% and nice rank stability up to 84%.

6 Characterizing the Generalization Barrier Words

6.1 Part-of-Speech distribution

In this part, we summarize the distribution of the detected generalization barrier words with respect to their Part-of-Speech (POS) tags. In order to consider the subword segments, we first use a POS tagger to label on the BPE-restored corpus, and then map the non-subword segments to the corresponding POS tags while the subword segments to a special tag named BPE, so that we can readily measure the ratio of subwords. The summary statistics are shown in Table 2. To compare with the natural distribution of all the words over POS, we also demonstrate them together with the detected generalization barrier words at the **base** column.

For both Chinese and English source inputs, barrier words are pervasive across all POS categories, since there is no significant difference from the base distribution. Note that, functional words like

Task	Word cat.	k=10%	k=20%	k=30%
Zh⇒En	Random	8.39%	17.04%	26.93%
	Frequency	8.10%	18.27%	26.91%
	Entropy	7.58%	18.42%	28.53%
	Exception	8.19%	17.14%	27.49%
En⇒Zh	Random	7.77%	17.60%	27.32%
	Frequency	8.57%	17.37%	25.29%
	Entropy	8.85%	18.71%	29.97%
	Exception	8.18%	17.96%	26.79%

Table 3: The overlap@k% metric with respect to different types of troublesome word statistics which due not utilize real decoding.

preposition and punctuation increase the most (with 3 $^+$) over the base. For English source, BPE is less tended to be barriers which indicate the benefit of subword-based segmentation. And for content words like noun and proper noun, they tend to be relatively less ambiguous and less context dependent thus tend to cause less problems.

6.2 Comparing to other source word categorizations

In this part, we compare the detected generalization barrier words with other source word categorizations: **a)** low-frequency words; **b)** high translation entropy words (Zhao et al., 2019); and **c)** exception words (Zhao et al., 2018). Words in a) are commonly said to cause generalization error, while words in b) and c) are dubbed as under-translated and troublesome words respectively according to the papers. Here, we want to know whether those probable trouble makers are barrier words who on average cause the most performance degradation?

Since a) - c) all use global statistics for each word $v \in \mathcal{V}$, to compare with the generalization barriers annotated with local risk, for each unseen input x, we also use x_i's global statistical clue to annotate itself in this local context so that overlap@k% can be used for comparison. The statistics are denoted as $1/freq(v)$, $te(v)$, $er(v)$ for inverse frequency, translation entropy and exception rate. Translation entropy of v is obtained through estimating the lexical translation probability $\phi(w|v)$ and compute the entropy of this distribution among all $w \in \mathcal{V}'$ (target vocabulary). Exception rate of a word v is calculated through the ratio between the number of exception alignment according to certain exception condition and the total number of alignment of v across the training corpus, $\frac{M^v}{N^v}$. Detailed introduction of the trouble makers is in Appendix A.3.

Table 3 demonstrates the overlap@k% values for Zh⇒En and En⇒Zh. The **random** row shows the metric values if we randomly choose an order of the source words. It is obvious that all categorizations are very close to random, with Entropy slightly better than random, which indicates our generalization barrier words that rely on statistics from inference-aware counterfactuals are very different from other source-side word categories. This highlights the novelty of such phenomenon, and implies the importance of studying generalization with explicit inference under consideration.

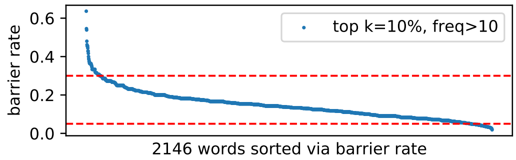


Figure 4: The distribution of barrier rate across words with context count larger than 10 on NIST 03-06.

6.3 Context sensitivity of barriers

In this section, we try to aggregate local statistics to obtain certain global understanding: is it possible that some words are prone to be generalization barriers in a context-agnostic way or the reverse. We aggregate the top-k% words in each test input and calculate their count. Specifically, if we assume that one appearance of a word roughly represents a context, we can calculate the probability of certain detected barrier word of being an universal barrier according to the following barrier rate:

$$p(v) = \frac{\sum_i \mathrm{Count}\Big(v|\mathrm{is\_Barrier}(v) \wedge v \in \mathrm{x}^i\Big)}{\sum_i \mathrm{Count}\Big(v|v \in \mathrm{x}^i\Big)} \quad (5)$$

We then summarize the distribution of each word's barrier rate in Figure 4. The two horizontal dashed red lines are 0.3 and 0.05, indicating *relatively* highly context-agnostic and context-sensitive respectively. As you can see, there are few context-agnostic barriers and most of the barrier words are very sensitive to context, indicating the necessity of mining large-scale training data with abundant contexts (Schwenk et al., 2019a,b). To obtain an intuition of the least/most context-sensitive barriers, we list some of them in Table 4. We can find that the least context-sensitive barriers are mostly high-frequency function words (e.g. punctuation) which tend to appear in all kinds of context; meanwhile the most context-sensitive barriers can be very frequently used nouns having low contextual-

		$p(v)$
Least sensitive	'(', '?', 'let', 'actually', 'regarding', 'forth', '2006', 'impact', 'entire', 'google', 'dalai'	>0.3
Most sensitive	'management', 'economy', 'health', 'finacial', 'help', 'technology', 'level', 'service'	<0.05
Not barriers	'on@@', 'clear', 'annual', 'base', 'town', 'leadership', 'v@@', 'confidence', 'television'	0.0

Table 4: A set of detected barrier words that are least/most context-sensitive on NIST 03-06 En⇒Zh.

Arch. pair	**k=10%**	**k=20%**	**k=30%**
random-random	18.61%	36.46%	50.12%
san-fconv	28.65%	45.03%	58.42%
san-rnn	25.46%	43.73%	57.68%
fconv-rnn	27.64%	45.52%	58.85%

Table 5: The overlap@k% statistics with respect to different architectural choices (on NIST03 Zh⇒En).

Task	**Candidates**	**Oracle ↑**	**Coverage ↑**	**Diversity ↓**
Zh⇒En	original	39.40	78.22	61.78
	ours	42.78 (+3.38)	83.88 (+5.66)	57.21 (-4.75)
En⇒Zh	original	32.31	72.10	59.98
	ours	37.48 (+5.17)	79.62 (+7.52)	52.72 (-7.26)

Table 6: The comparison of various properties of the reranking candidates generated between top-k decoding over the original input (original) and top-1 decoding over the randomly edited inputs (ours).

ity (Ethayarajh, 2019), while BPE token tends to be less a barrier.

6.4 Relation to training data

In this section, we ask the question: although the barrier words are very context-sensitive, do they tend to be model-agnostic and caused largely by what data the model is trained upon? We train three representative model architectures rnn, cnn, san and compute their pair-wise barrier precision against a random baseline. Table 5 shows the overlap between different architectures. Although the overlap is still relatively low, all of them are consistently higher than the random baseline, indicating that the same training data does contribute to the learning towards similar barrier words. However, there are many other factors that we do not control in our experiments, for example, the order of the training batches, aka learning curriculums are not the same across different archs, which may still contribute to the sensitivity of detected barriers.

7 Potential Usage: Reranking

The previous sections empirically show that it is possible to improve translation quality by modifying some barrier words within the input, and these findings motivate us to present one potential usage of them in an automatic way through reranking (Yee et al., 2019). Since in the re-ranking process the reference translations are not available, we can not calculate the truncated mean for a word any more. Therefore, we firstly enumerate all words within the input and randomly edit each of them; then we perform top-1 decoding for all edited inputs to obtain re-ranking hypotheses. [2] Table 6 shows that the hypotheses collected from top-1 decoding over the edited inputs deliver higher oracle performance, better translation recall and diversity than top-k candidates over the original single input, which is currently the common wisdom of reranking for NMT. Actually, we find that, the usual top-k candidates are very similar to each other and the oracle translation seems to be a paraphrased version of the highest model-scored one which might be very hard for the reranking model to pick up, instead the candidates generated by editing barriers can recall the actual incorrectly or un-translated parts of meaning of the source. Details for the measures are introduced in Appendix A.4.

[2]This is similar to Algorithm 1 with S = *Uniform* except that it collects hypotheses without calculating truncated mean.

8 Conclusion and Future Work

In this paper, we identify and define a new phenomenon in NMT named generalization barrier as a *media* for understanding behavior of NMT model. Simple approximation methods are investigated to efficiently detect such generalization barrier words. After large-scale detection on held-out test sets, we find that barrier words are very context-dependent, highly related to the training corpus and model-sensitive. And they are very different from previously identified trouble makers in the source side. These analyses somehow prove the complexity of current NMT model and efforts or theories for better understanding them in terms of its generalization ability should continue. Future work involves fundamental **causal analysis** of the emergence of such phenomenon *intrinsically* through the lens of the learned representation and representation confounding effect (Li et al., 2019) or *extrinsically* through compositionality study of the input.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *ArXiv*, abs/1606.06565.

Collin Burns, Jesse Thomason, and Wesley Tansey. 2019. Interpreting black box models with statistical guarantees. *arXiv preprint arXiv:1904.00045*.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining image classifiers by counterfactual generation. In *ICLR*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *ACL*.

Hyunsun Choi, Eric Jang, and Alexander A. Alemi. 2018. Waic, but why? generative ensembles for robust anomaly detection. In *arXiv*.

Hal Daumé III. 2006. *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California, Los Angeles, CA.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2376–2384, Long Beach, California, USA. PMLR.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 953–962, Hong Kong, China. Association for Computational Linguistics.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. 2019. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4120–4133, Hong Kong, China. Association for Computational Linguistics.

Wenqiang Lei, Weiwen Xu, Ai Ti Aw, Yuanxin Xiang, and Tat Seng Chua. 2019. Revisit automatic error detection for wrong and missing translation – a supervised approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 942–952, Hong Kong, China. Association for Computational Linguistics.

Ke Li, Tianhao Zhang, and Jitendra Malik. 2019. Approximate feature collisions in neural nets. In *Advances in Neural Information Processing Systems*, pages 15816–15824.

Shiyu Liang, Yixuan Li, and R. Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *International Workshop on Spoken Language Translation*.

Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497.

Jirí Mazurek. 2011. Evaluation of ranking similarity in ordinal ranking problems. In *Acta academica karviniensia*.

Behrang Mohit and Rebecca Hwa. 2007. Localization of difficult-to-translate phrases. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 248–255, Prague, Czech Republic. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*, 1st edition. Basic Books, Inc., New York, NY, USA.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems 32*, pages 14680–14691. Curran Associates, Inc.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019a. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019b. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, Florence, Italy. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5700–5705, Hong Kong, China. Association for Computational Linguistics.

Yang Zhao, Jiajun Zhang, Zhongjun He, Chengqing Zong, and Hua Wu. 2018. Addressing troublesome words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 391–400.

Yang Zhao, Jiajun Zhang, Chengqing Zong, Zhongjun He, Hua Wu, et al. 2019. Addressing the under-translation problem from the entropy perspective. In *Vol 33 (2019): Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

A Appendices

A.1 Mean performance

Task	Model	Train	Dev.	Test Avg.
Zh⇒En	rnn	35.02	41.02	37.73
	fconv	40.02	45.58	43.04
	san	38.46	47.85	45.17
En⇒Zh	rnn	39.12	22.57	16.61
	fconv	40.91	24.96	18.43
	san	41.67	26.31	19.50

Table 7: The average sense generalization performance results on NIST benchmark measured by BLEU; note that here **Train** is measured through single reference while **Dev.** is measured by four references for the Zh⇒En task, so for rnn, **Dev.** can surpass **Train**.

A.2 Evaluation metrics

overlap@k% The first metric we use for evaluating the accuracy of the estimated risk is based on the overlap@k metric (Dong et al., 2018). Since each source word x_i is annotated with a risk r_i via exactly or approximately generating couterfactuals. The risks then induce a ranking among the source words. According to our Definition 3.2, the top-k% risky words are treated as generalization barrier words. So given two rankings of the same input, we can choose their top-k% risky words and measure how they overlap with each other. Formally, given two ranked list of words of the input x based on two list of risks, τ_1 and τ_2 are their top-k risky words, the overlap@k% metric is as follows:

$$\text{overlap@}k\% = \frac{\tau_1 \cap \tau_2}{k\% \cdot l}, \tag{6}$$

where l is the length of the source input.

Kendall's coefficient concordance The second metric for evaluting rank stability (variance) is called Kendall's coefficient of concordance (Mazurek, 2011). It is computed through the following formula:

$$W = \frac{\sum_{i=1}^{n} X_i^2 - \frac{(\sum_i^n X_i)^2}{n}}{\frac{1}{12} \cdot k^2 \cdot (n^3 - n)}, \tag{7}$$

where k is the number of rankings and n the number of objects. In our setting, k is 25 corresponding to the 25 repeats of the simulation and n is the source sentence length corresponding to the length of ranks on all the source words.

A.3 Definition of troublesome words

In Section 6.2, we measure the similarity between our identified generalization barrier words and previously proposed under-translated words (Zhao et al., 2019) and troublesome words (Zhao et al., 2018). Here, we give a detailed introduction to the definition of them.

Under-translated words The under-translated word $v \in \mathcal{V}^s$ (Zhao et al., 2019) is defined as the word with its translation entropy larger than certain threshold. Each word's translation entropy is calculated from its translation probabilities $\phi(w|v)$ which are count-based estimated from word alignments of the training set obtained through certain statistical word aligner, e.g. fast_align (Dyer et al., 2013). That is, for each $v \in \mathcal{V}^s$, $te(v) = \sum_w -\phi(w|v) \cdot \log \phi(w|v)$, where $w \in \mathcal{V}'$. So we can use $te(v)$ of each word to annotate each source sentence with every word with a global risk.

Troublesome words The troublesome word v (Zhao et al., 2018) is defined as word that satisfies certain exception condition, which is measured through an exception rate $er(v) = \frac{M^v}{N^v}$. Here, N^v is the number of alignment pair (v, w) for any $w \in \mathcal{V}'$, across the whole corpus obtained as well with fast_align; M^v is the number of exception alignment pair where w has violated certain conditions. Zhao et al. (2018) proposes three exception conditions which result in similar performance, so here we use only one of them for experiment. That is, the word probability $P_{\hat{\theta}}(\mathrm{y}_t = w|\mathrm{y}_{<t}, \mathrm{x})$ falls below certain threshold p_0. The same with the under-translated word, we use $er(v)$ to label each source word.

A.4 Measures for evaluating the re-ranking candidates

In Section 7, we use three measures to characterize the candidates generated by top-1 beam search from several randomly edited sources via barrier words and commonly used top-k beam search results from the original source input. Here, we give a detailed description of those measures. We denote the hypo candidates generated from source-editing top-1 beam search and top-k beam search as $\mathcal{C}_1$ and $\mathcal{C}_2$. To be fair, the two collections of hypo candidates have same size, that is $|\mathcal{C}_|| = |\mathcal{C}_2|$.

Oracle Given the reference y^*, a set of candidates $\mathcal{C}_i$ $(i \in \{1, 2\})$, the oracle value of $\mathcal{C}_i$ is:

$$\mathcal{O}(\mathcal{C}_i, \mathrm{y}^*) = \max_{\hat{\mathrm{y}} \in \mathcal{C}_i} \mathrm{BLEU}(\hat{\mathrm{y}}, \mathrm{y}^*), \tag{8}$$

where the function BLEU denotes the sentence-level smoothed BLEU (Lin and Och, 2004) in all our experiments. The larger the oracle value is, the better the candidates are.

Coverage Given the reference y*, a set of candidates $\mathcal{C}_i$ ($i \in \{1, 2\}$), the coverage value of $\mathcal{C}_i$ is:

$$\mathcal{C}(\mathcal{C}_i, \mathrm{y}^*) = \frac{\text{1-Gram}(\mathrm{y}^*) \cap \cup_{\hat{\mathrm{y}} \in \mathcal{C}_i} \text{1-Gram}(\hat{\mathrm{y}})}{\text{1-Gram}(\mathrm{y}^*)}, \tag{9}$$

where 1-Gram$(\cdot)$ denotes the different 1-grams of the sentence $\cdot$. The layer the coverage value is, the better the candidates are.

Diversity Given a set of candidates $\mathcal{C}_i$ ($i \in \{1, 2\}$), the diversity value of $\mathcal{C}_i$ is:

$$\mathcal{D}(\mathcal{C}_i) = \frac{1}{|\mathcal{C}_i| * (|\mathcal{C}_i - 1|)} \sum_{\hat{\mathrm{y}} \in \mathcal{C}_i, \hat{\mathrm{y}}' \in \mathcal{C}_i} \text{BLEU}(\hat{\mathrm{y}}, \hat{\mathrm{y}}'), \tag{10}$$

where $\hat{\mathrm{y}} \neq \hat{\mathrm{y}}'$. That is we use the sentence-level smoothed BLEU for comparing the difference between any two candidates and average them all. So the smaller the diversity value is, the better the candidates are.

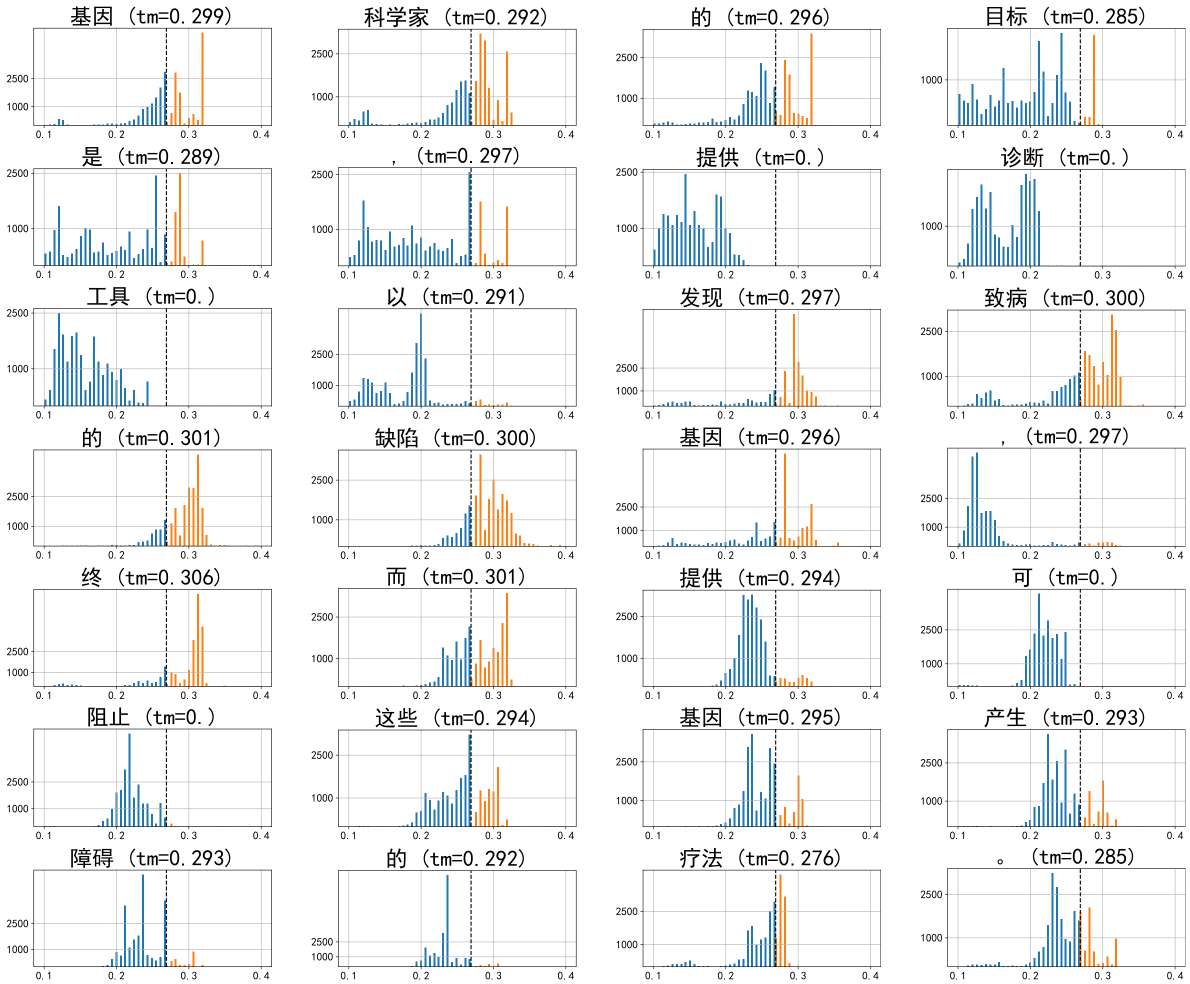


Figure 5: The all 28 words' histogram via exhaustive editing each words