# SYSTRAN @ WMT24 Non-Repetitive Translation Task

**Marko Avila** and **Josep Crego**

SYSTRAN by ChapsVision
5 rue Feydeau,
75002 Paris, France
{mavila,jcrego}@chapsvision.com

## Abstract

Many contemporary NLP systems rely on neural decoders for text generation, which demonstrate an impressive ability to generate text approaching human fluency levels. However, in the case of neural machine translation networks, they often grapple with the production of repetitive content, also known as repetitive diction or word repetition, an aspect they weren't explicitly trained to address. While not inherently negative, this repetition can make writing seem monotonous or awkward if not used intentionally for emphasis or stylistic purposes. This paper presents our submission to the WMT 2024 Non-Repetitive Translation Task, for which we adopt a repetition penalty method applied at learning inspired by the principles of label smoothing. No additional work is needed at inference time. We modify the ground-truth distribution to steer the model towards discouraging repetitions. Experiments show the ability of the proposed methods in reducing repetitions within neural machine translation engines, without compromising efficiency or translation quality.

## 1 Introduction

The Non-Repetitive Translation Task of the ninth Conference on Machine Translation (WMT24) focuses on lexical choice in machine translation, especially choice regarding repeated words in a source sentence. Generally, the repetition of the same words can create a monotonous or awkward impression in English, and it should be appropriately avoided. Typical workarounds in monolingual writing are to

1) remove redundant terms if possible (reduction) or

2) use alternative words such as synonyms as substitutes (substitution).

These techniques are also observed in human translations. The goal of this task is to study how these techniques can be incorporated into machine translation systems to enrich lexical choice capabilities. From a practical standpoint, such capability would be important, for example, in news production, where high quality text that goes beyond robotic word-by-word translation is required.

In addition, repetitions do not always have a negative impact on readability. Without aiming to be exhaustive : i) repetitions play a role when summarizing information or reinforcing a concept ; ii) common expressions are formed using word repetitions, and altering them to eliminate repetition would alter their intended meaning ; iii) in highly specialized domains, expressions convey precise meanings that disallow being reformulated. The following examples illustrate these observations :

i) once closed, the door stays closed

ii) over and over ; to be or not to be ; step by step

iii) the congenital muscular dystrophy in newborns presenting with muscular hypotonia

As previously introduced, finding suitable alternatives without altering the meaning of a sentence can be a challenging task.

Participants are required to control a machine translation system using reduction or substitution so that it does not output the same words for certain repeated words in a source sentence. The translation direction is Japanese to English.

## 2 Related Work

The fluency levels achieved by LLMs are widely acknowledged to be high, primarily owing to the extensive availability of monolingual datasets, which surpasses that of standard neural machine translation (NMT) models trained solely on parallel texts. To the best of our knowledge, no dedicated research has been conducted on addressing the repetition issue tackled in this work within NMT systems.



*Proceedings of the Ninth Conference on Machine Translation*, pages 1056–1062

Closely related, Welleck et al. (2019) describe a method to train neural language models that in addition to maximizing likelihood to model the overall sequence probability distribution, also includes an unlikelihood term in the loss function to correct known biases such as repeated tokens. Li et al. (2020) use the same approach to control copy effect and repetitions observed in dialogue tasks. Su et al. (2022) present a contrastive solution to encourage diversity while maintaining coherence in the generated text.

Various studies have addressed diversity in neural MT systems, which is a closely related topic. Sampling predictions from the output distribution can be an effective decoding strategy for back-translation, as described by Edunov et al. (2018), or sampling from less likely tokens Holtzman et al. (2020). Results show that such techniques enlarge diversity and richness of the generated translations when compared to data generated by beam or greedy search, but introduce semantic inconsistency in translations. In Lin et al. (2022) is proposed a multi-candidate optimization framework for augmenting diversity. The authors propose to guide an NMT model to learn more diverse translations from its candidate translations based on reinforcement learning. During training, the model generates multiple candidate translations, of which rewards are quantified according to their diversity and quality.

A different approach attempts to condition the decoding procedure with diverse signals. Typically, Shu et al. (2019) use syntactic codes to condition the translation process. Lachaux et al. (2020) replace the syntactic codes with latent domain variables derived from target sentences. Similarly, Schioppa et al. (2021) use prefix-based control tokens and vector-based interventions for controlling output translations from a NMT system. In the context of paraphrase generation Vahtola et al. (2023) propose a translation-based guided paraphrase generation model that learns useful features for promoting surface form variation in generated paraphrases.

## 3 Adjusting the ground-truth distribution

Throughout the training process, at every time-step $t$, neural machine translation networks generate predictions over the target-side vocabulary based on the input $x$ and previous predictions $y_{<t}$:

$$p_t^i = p(y_t^i|x, y_{<t}),\ i \in [1, ..., V]$$

where $V$ indicates the size of the target vocabulary.

The loss function evaluates the neural network's capacity to model the training data by comparing its predictions to a reference target vector $r = [r_1, r_2, ..., r_T]$, where $T$ denotes the sequence length. This loss is utilized to update the network's parameters, aiming to minimize the observed error in the model. The loss at time-step $t$ is usually computed as the cross-entropy between the model predictions $p_t = [p_t^1, ..., p_t^V]$ and the ground-truth distribution $q_t = [q_t^1, ..., q_t^V]$:

$$_t = -\sum_{i=1}^{V} q_t^i \, log(p_t^i) \quad (1)$$

Note that the vector $q_t$ is a one-hot encoding representation of $r_t$, with all entries set to 0 except for the token indicated by $r_t$, which is set to 1. Addressing the over-fitting risk illustrated by the previous $q_t$ distribution, label smoothing Szegedy et al. (2015); Müller et al. (2019) (LS) is widely employed to achieve a smoother distribution:

$$q_t^{\epsilon LS} = (1 - \epsilon)q_t + \frac{\epsilon}{V} \quad (2)$$

with $\epsilon$ being a commonly small hyper-parameter.[1]

| t | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| r | I | like | cookies | and | cookies | . |

| | | | | | | |
|---|---|---|---|---|---|---|
| . | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 |
| and | 0 | 0 | 0 | 0 | 0 | 0 |
| like | 0 | 0 | 0 | 0 | 0 | 0 |
| cookies | 0 | 0 | 0 | 0 | 1 | 0 |

Figure 1: Matrix for the ground-truth $r$ ='*I like cookies and cookies.*'. Rows $t$ and $r$ represent respectively the time-step and the corresponding ground-truth token. A reduced model vocabulary (matrix rows) is used to facilitate reading.

LS can be interpreted as penalizing the probability of the ground-truth class by a factor of $1 - \epsilon$, while evenly distributing the removed probability mass among all classes, $\epsilon/V$. Building upon a strategy akin to label smoothing, we make additional adjustments to the ground-truth distribution and reduce the likelihood of repeated tokens, with the

[1] $\epsilon = 0$ yields the initial distribution $q_t$, whereas $\epsilon = 1$ implies a uniform distribution.

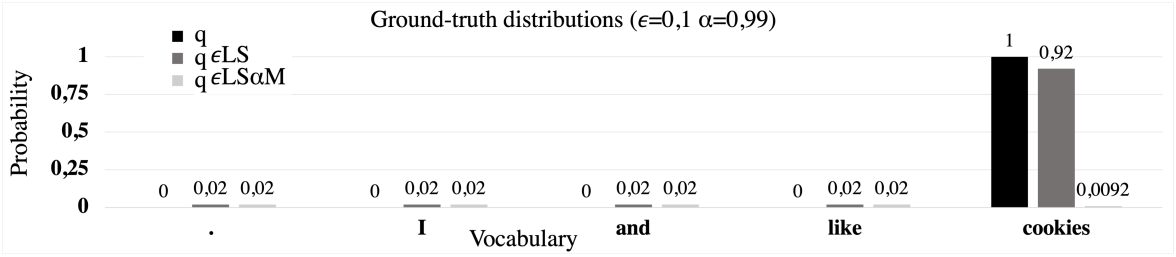


Figure 2: Ground-truth distributions for the $5^{th}$ time-step of our example: the original one-hot encoding $q$; adjusted with label smoothing $q^{\epsilon LS}$; and further adjusted with repetitions $q^{\epsilon L\alpha}$.

goal of enabling the model to learn to predict repetitions with lower probability. We introduce a matrix, denoted as $_{V\times T}$, which indicates whether the ground-truth token $r_t$ is also present in the preceding time-steps.[2] Figure 1 illustrates an example of matrix with ground-truth *I like cookies and cookies.* as translation of the Japanese sentence クッキーとビスケットが好き with a model vocabulary of 5 tokens (matrix rows). Both Japanese terms クッキー *[cookies]* and ビスケット *[biscuits]* are correctly translated into English as *cookies*, yet this choice clearly reduces the fluency and clarity of the translation. As it can be seen, only $_{[i=5,t=5]}$ is set to 1 since only $r_5$ ='*cookies*' occurs in a preceding time-step ($t = 3$).

We consequently update the ground-truth distribution following:

$$q_t^{\epsilon LS\alpha} = (1-\epsilon)(1-\alpha_t)\, q_t + \frac{\epsilon}{V} \qquad (3)$$

where $\alpha$ is a hyper-parameter, and $\alpha$ is used as a penalty, much like $\epsilon$ in the case of LS. Note that only the label smoothing probabilities discounted are distributed among all classes. As a result, time-steps with repeated tokens (such as $t = 5$ in our example) do not constitute proper probability distributions, as their sum does not add to 1. Figure 2 illustrates ground-truth distributions for our example at time-step $t = 5$: the original one-hot encoding $q$; the original distribution adjusted using label smoothing $q^{\epsilon LS}$, and further adjusted using repetitions $q^{\epsilon LS\alpha}$.[3] A significant challenge with the aforementioned techniques that modify $q$ distribution with repetitions is their limited impact on the training process, primarily caused by the scarcity of repeated tokens in datasets. In the following section, we present alternative approaches to address this challenge.

## 4 Gathering Examples with Repetitions

As previously depicted, our intention is to instruct the model to minimize certain repetitions while preserving others deemed necessary for an accurate translation. To achieve this, we must compile a relatively large dataset of examples that demonstrate this behavior to the model. We initially focus on repetitions of content words such as *nouns*, *adjectives*, *verbs*, and *adverbs*. Function words, which serve a distinct grammatical role in a sentence, are excluded from this analysis. Current MT networks reliably generate these words based on their understanding of grammatical correctness.

We back-translate the Japanese side of the JiJi corpus (further detailed in Section 5.1) into English and annotate word (or sequence) repetitions of content words based on automatic morpho-syntactic annotations performed by Spacy[4]. We employ word-alignments between Japanese and English words performed by the Giza++ Och and Ney (2003) toolkit[5] in order to consider only repetitions of English content words aligned to Japanese content words (Verbs, Nouns, Adjectives and Adverbs). The resulting set of examples with repetitions from src/tgt training pairs will be regarded as instances that the model needs to learn to discourage. Consequently, we utilize them for training after annotating the repeated target words in their respective matrices.

It's worth noting that the presented approach does not require any alterations to the network architecture and maintains the same training and inference efficiency.

[2]Note that repetitions are computed over words while matrix refers to tokens $r \in V$ for each time-step $t \in T$.

[3]As previously discussed, distribution $q^{\epsilon LS\alpha}$ does not form a proper distribution since probabilities do not add to 1 ($0,02+0,02+0,02+0,02+0,0092 = 0,0892$). We leave for future experiments the normalization of the output scores in order to allow for a valid probability distribution.

[4]https://spacy.io/

[5]https://github.com/moses-smt/giza-pp.

| | |
|---|---|
| Jap | JEMAの担当者は白物家電について、「機能を絞った低価格製品、高価格な高機能製品とも好調だ」と述べている。 |
| Eng | "Shipments have been robust for both low-priced models with reduced functions and expensive high-spec products," a JEMA official said. |
| Jap | JEMAの担当者は白物家電について、「<target id=0 ref=0 type=s>機能<\target>を絞った低価格<target id=1 ref=0 type=s>製品<\target>、高価格な高<target id=0 ref=1 type=s>機能<><target id=1 ref=1 type=s>製品<\target>とも好調だ」と述べている。 |
| Eng | "Shipments have been robust for both low-priced <target id=1 ref=0 type=s>models<\target> with reduced <target id=0 ref=0 type=s>functions<\target> and expensive <target id=0 ref=1 type=s>high-spec<\target> <target id=1 ref=1 type=s>products<\target>," a JEMA official said. |

Table 1: example of Japanese-English translation: raw translation is shown at the top, and the tagged translation to annotate repetitions is shown at the bottom.

## 5 Experimental Framework

### 5.1 Datasets

We evaluate the proposed methods in a Japanese-to-English translation task. Thus, we utilize Japanese-English parallel corpora freely obtained from the WMT24 for Non-Repetitive Translation Task website[6]. The corpus is compiled by Jiji Press Ltd in collaboration with the National Institute of Information and Communication Technology (NICT) with various categories, including politics, economy, nation, business, markets, sports, etc., for use in machine translation, in particular for previous the Workshop on Asian Translation (WAT)[7].

Table 2 presents various statistics of the corpora used in this work, including the total number of sentences, vocabularies, words, and average sentence length. Statistics are computed after performing a light tokenization aiming to split-off punctuation. For testing, we use the supplied Japanese-English datasets made available by the task organizers.

| Lang | #Sents | #Vocab | Words | Length |
|---|---|---|---|---|
| *Training-set* | | | | |
| Jap | 200k | 49$K$ | 6.9M | 4.46 |
| Eng | | 118K | 4.5M | 24.64 |
| *Repetition-set* | | | | |
| Jap | 470 | 3,297 | 23,472 | 4.22 |
| Eng | | 4,341 | 13,814 | 11.91 |

Table 2: Corpora statistics. M and K stand for millions and thousands respectively.

Due to the poor alignment quality of the Japanese-English parallel sentences present in the provided dataset (sentence pairs are coupled using an automatic cross-lingual sentence similarity score) we decided to back-translate the English side using an in-house English-Japanese model. Then, using the resulting Japanese[8]-English dataset we fine-tune our baseline Japanese-English model.

In addition, we use a test set of repetitions also provided by the challenge, consisting of reference English machine translations and their corresponding Japanese machine translations that include at least one word repeated on the target (English) side for a more nuanced analysis of repetition. Among the files corresponding to the test datasets are those containing tagged files in which repeated words and their translations in each sentence pair are marked with tags <*target*> and <*/target*>. Marked words indicate that they are evaluated repetitions. Three labels, 'id', 'ref' and 'type' are embedded within the tags. Table 1 illustrates an example, where:

**id** indicates IDs of repated words. In the above example, two tagged repeated words are included, i.e., 機能 (id=0) and 製品 (id=1). The number of instances including multiple id's, such as the above example, are limited.

**ref** indicates IDs of pairs of source/target words, such as 製品/*models* (id=1, ref=0) and 製品/*products* (id=1, ref=1).

**type** indicates whether they are substituted (*s*) or reduced (*r*).

The *Repetition-set* is mainly used to evaluate the performance of our models in handling repetition problems, as well as to assess overall translation accuracy.

[6] https://www2.statmt.org/wmt24/non-repetitive-translation-task.html

[7] https://lotus.kuee.kyoto-u.ac.jp/WAT/

[8] Back-translated from English.

### 5.2 NMT Models

Our NMT model is built using an in-house implementation of the state-of-the-art Transformer architecture Vaswani et al. (2017). Details of the network hyper-parameters emplooy for training are given in Table 3.

| | |
|---|---|
| size of word embedding | 512 |
| size of hidden layers | 512 |
| size of inner feed forward layer | 2,048 |
| number of heads | 8 |
| number of layers | 6 |
| batch size | 4,000 (tokens) |
| batch accumulation | 25 (batches) |

Table 3: Network hyperparameters.

For optimization work we use the lazy Adam algorithm Kingma and Ba (2014). We set warmup steps to $4,000$ and update learning rate for every $8$ iterations. All models are trained using a single NVIDIA V100 GPU.

We limit the source and target sentence lengths to $150$ tokens based on BPE Sennrich et al. (2016) preprocessing. A total of $28K$ BPE merge operations are separately computed for each language. We finally use a joint Japanese and English vocabulary of $58K$ tokens. In inference we use a beam size of $5$.

Our *baseline* English-to-Japanese model is trained during more than 3 million iterations using all the parallel data available in the Opus website[9].

## 6 Results

To evaluate the method presented in this paper we consider the previous *baseline* model that we update with 15K additional iterations for two different configurations of the ground-truth distribution:

$q^{\epsilon LS}$ follows the same configuration than our *baseline* model with label smoothing set to $\epsilon = 0.1$.

$q^{\epsilon LS\alpha}$ further penalizes the ground-truth distribution with repetition penalties as detailed in Section 3 with $\epsilon = 0.1$ and for different values of $\alpha$.

Note that for both configurations, we use the same training corpus detailed in Table 2 (*Training-set*).

We also assess the effectiveness of two large language models (LLM) with translation capabilities to overcome the repetition issue:

$GPT3.5$ consists of the *GPT3.5-turbo* version of the OpenAI LLM. Built upon the Generative Pre-trained Transformer architecture Radford and Sutskever (2018) which employs only a transformer decoder. Following an auto-regressive approach, the model ensures that the generated text maintains coherence and relevance to the context provided by the input text. Translations are conducted using the OpenAI API, while emphasizing the importance of minimizing word repetitions through the provided prompt: *Translate the following text from English to Japanese, ensuring that the translated output maintains coherence and fluency while minimizing the repetition of words or phrases. Pay attention to using synonyms, varied sentence structures, and appropriate linguistic devices to enhance the overall quality of the translation. Feel free to creatively adapt the language to achieve a natural and engaging tone in the target language. I want you to only reply the translation, do not write explanations*.

$NLLB$ is a family of machine translation models based on the Transformer encoder-decoder architecture, enabling translation between any of the 202 language varieties NLLB Team et al. (2022). We use the *nllb-200-distilled-600M*[10] version and perform translations with the efficient CTranslate2[11] inference toolkit.

To evaluate the presented methods, we report `BLEU` results computed by `sacrebleu`[12] Post (2018) respectively over test sets. We also report the number of word repetitions that hinder fluency, *Degrading*, after a human evaluation performed on translation hypotheses. Table 4 summarize results obtained by different system configurations.

Models fine-tuned from the *baseline* network exhibit nearly identical quality scores across the *test* set. This suggests that training with the method presented to adjust the ground-truth distribution does not compromise translation quality. On the contrary, unlike Configuration $q^{\epsilon LS}$, Configurations $q^{\epsilon LS\alpha}$ demonstrate a significant decrease in the number of repetitions that degrade fluency over the *Repetition-set*, while retaining most of the acceptable repetitions in the translated output.

[9] https://opus.nlpl.eu/

[10] https://huggingface.co/facebook/nllb-200-distilled-600M

[11] https://github.com/OpenNMT/CTranslate2

[12] https://github.com/mjpost/sacrebleu

Results from both LLMs demonstrate a reduced number of repetitions, suggesting an elevated level of diversity and fluency of such models. However, the translation quality scores of LLMs do not align with those achieved by the models presented in this study in either of the test sets, especially translations obtained by GPT-3.5. These findings are consistent with those presented by Bawden and Yvon (2023) where the authors note the challenge of controlling translations performed by BLOOM[13], a multilingual LLM.

| Configuration | BLEU | *Degrading* |
|---|---|---|
| $q^{\epsilon LS}$ | 28.41 | 77 |
| $q^{\epsilon LS\alpha}, 1-\alpha = 10^{-6}$ | 28.91 | 60 |
| $GPT3.5$ | 19.29 | 64 |
| $NLLB$ | 16.12 | 74 |

Table 4: Translation accuracy results and number of repetitions present in translations performed by models under different configurations. $\epsilon$ is always set to 0.1.

## 7 Conclusions and Further Work

We presented SYSTRAN submission to the WMT24 Non-Repetitive Translation Task. Our NMT systems introduce a method to reduce the occurrence of repetitions in translation hypotheses, which significantly affects the readability of the generated texts. The method is solely implemented during fine-tuning at the conclusion of the training phase, without any modifications to the inference process. Experiments indicate the ability of our proposed methods in reducing the repetition problem.

We aim to further study the impact of the ratio between the number of reference sentences and synthetic translations that include repetitions during the training process. Additionally, we plan to analyze the influence of the distance (measured in number of words) between repetitions and explore the possibility of replacing the binary penalty in matrix  with a softer approach.

## Acknowledgements

The work presented in this paper was supported by the EU Horizon 2020 Programme for TRACE project. We also would like to thank the thorough evaluation and valuable insights provided by the reviewers.

[13] https://huggingface.co/bigscience/bloom

## References

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Marie-Anne Lachaux, Armand Joulin, and Guillaume Lample. 2020. Target conditioning for one-to-many generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2853–2862, Online. Association for Computational Linguistics.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

Huan Lin, Baosong Yang, Liang Yao, Dayiheng Liu, Haibo Zhang, Jun Xie, Min Zhang, and Jinsong Su. 2022. Bridging the gap between training and inference: Multi-candidate optimization for diverse neural machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2622–2632, Seattle, United States. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. *When Does Label Smoothing Help?* Curran Associates Inc., Red Hook, NY, USA.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp

Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Narasimhan Karthic Salimans Tim Radford, Alec and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical Report*.

Andrea Schioppa, David Vilar, Artem Sokolov, and Katja Filippova. 2021. Controlling machine translation for multiple attributes with additive interventions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6676–6696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, Florence, Italy. Association for Computational Linguistics.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Teemu Vahtola, Mathias Creutz, and Jrg Tiedemann. 2023. Guiding zero-shot paraphrase generation with fine-grained control tokens. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 323–337, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *ArXiv*, abs/1908.04319.