# Vega-MT: The JD Explore Academy Translation System for WMT22

**Changtong Zan**[ℜ,♭,🧸], **Keqin Peng**[♯,🧸], **Liang Ding**[ℜ,🧸], **Baopu Qiu**[♮], **Boan Liu**[♢], **Shwai He**[△]
**Qingyu Lu**[♡], **Zheng Zhang**[♢], **Chuang Liu**[♢], **Weifeng Liu**[♭], **Yibing Zhan**[ℜ], **Dacheng Tao**[ℜ]
[ℜ]JD Explore Academy, JD.com Inc.
[♭]China University of Petroleum (East China) [♯]Beihang University [♮]Nanjing University
[♢]Wuhan University [△]University of Electronic Science and Technology of China [♡]Southeast University
dingliang1@jd.com

## Abstract

We describe the JD Explore Academy's submission of the WMT 2022 shared task on general machine translation. We participated in all high-resource tracks and one medium-resource track, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En). **[Method]** We push the limit of our previous work – bidirectional training (Ding et al., 2021d) for translation by scaling up two main factors, *i.e.* language pairs and model sizes, namely the **Vega-MT** system. As for language pairs, we scale the "bidirectional" up to the "multidirectional" settings, covering all participating languages, to exploit the common knowledge across languages, and transfer them to the downstream bilingual tasks. As for model sizes, we scale the Transformer-BIG up to the extremely large model that owns nearly 4.7 Billion parameters, to fully enhance the model capacity for our Vega-MT. Also, we adopt the data augmentation strategies, *e.g.* cycle translation (Ding and Tao, 2019) for monolingual data, and bidirectional self-training (Ding and Tao, 2021) for bilingual and monolingual data, to comprehensively exploit the bilingual and monolingual data. To adapt our Vega-MT to the general domain test set, generalization tuning is designed. **[Results]** Based on the official automatic scores* of constrained systems, in terms of the SACREBLEU (Post, 2018) shown in Figure 1, we got the 1st place in {Zh-En (33.5), En-Zh (49.7), De-En (33.7), En-De (37.8), Cs-En (54.9), En-Cs (41.4) and En-Ru (32.7)}, 2nd place in {Ru-En (45.1) and Ja-En (25.6)}, and 3rd place in {En-Ja(41.5)}, respectively; W.R.T the **COMET** (Rei et al., 2020), we got the 1st place in {Zh-En (45.1), En-Zh (61.7), De-En (58.0), En-De (63.2), Cs-En (74.7), Ru-En (64.9), En-Ru (69.6) and En-Ja (65.1)}, 2nd place in {En-Cs (95.3) and Ja-En (40.6)}, respectively. Models will be released to facilitate the MT community through GitHub† and OmniForce Platform‡.

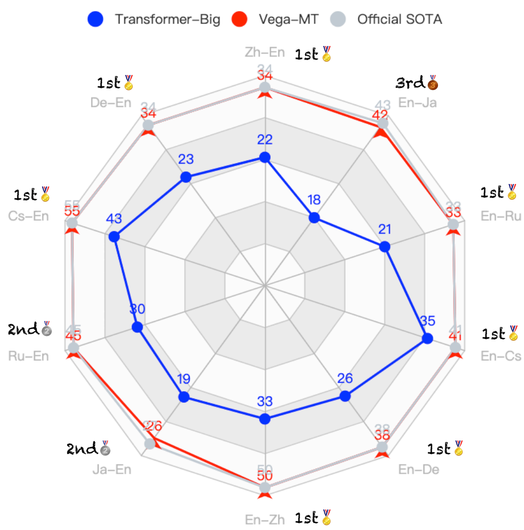


Figure 1: Vega-MT achieves 7 state-of-the-art BLEU points out of 10 high-resource translation tasks among all constrained systems, and significantly outperforms the competitive Transformer-BIG baselines.

## 1 Introduction

In this year's WMT general translation task, our Vega-MT translation team participated in 10 shared tasks, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En). We use the same model architectures, data strategies and corresponding techniques for all tasks.

🧸 Equal contribution. Work was done when Changtong and Keqin were interning at JD Explore Academy.

*https://github.com/wmt-conference/wmt22-news-systems/tree/main/scores

†https://github.com/JDEA-NLP/Vega-MT

‡OmniForce Platform will be launched by JD Explore Academy



*Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 411 - 422
December 7-8, 2022 ©2022 Association for Computational Linguistics

We aim to leverage the cross-lingual knowledge through pretraining (PT) to improve the high-resource downstream bilingual tasks. Although recent works (Song et al., 2019; Lewis et al., 2020; Liu et al., 2020b; Wang et al., 2022) attempt to leverage sequence-to-sequence PT for neural machine translation (NMT; Bahdanau et al., 2015a; Gehring et al., 2017; Vaswani et al., 2017a) by using a large amount of unlabeled (*i.e.* monolingual) data, Zan et al. (2022b) show that it usually fails to achieve notable gains (sometimes, even worse) on resource-rich NMT on par with their random-initialization counterpart, which is consistent with our preliminary experiments. Ding et al. (2021d) show that bidirectional pretrained model as initialization for downstream bilingual tasks could consistently achieve significantly better performance. It is natural to assume that scaling the "bidirectional" to the "multidirectional" setting with {1) *multilingual pretraining* and 2) *large enough model capacity*} could benefit the downstream resource-rich bilingual translations. Tran et al. (2021) and Lin et al. (2020) also provide empirical evidences to support our motivation of supervised multilingual pretraining. Different from Tran et al. (2021) that explores the effectiveness of multilingual training, we show that further tuning on the bilingual downstream task provide more in-domain knowledge and thus could gain better translation quality. Compared with Lin et al. (2020), our model do not require any alignment information during pretraining, which will consume more extra time and computation resources, making our strategy flexible to be applied to any language.

For model frameworks in §2.1, we tried autoregressive neural machine translation, including Transformer-BIG and -XL (Vaswani et al., 2017b), and non-autoregressive translation models (Gu et al., 2018), where the Transformer-XL is employed as the foundation model and autoregressive BIG and non-autoregressive models are used during augmenting. For the core training strategy of our Vega-MT, we cast the multilingual pretraining as foundation models in §2.2, including MULTI-DIRECTIONAL PRETRAINING (§2.2.1) and SPECIFIC-DIRECTIONAL FINETUNING (§2.2.2). For data augmentation strategies, we employ CYCLE TRANSLATION (§2.3.1) and BIDIRECTIONAL SELF-TRAINING (§2.3.2) for both monolingual and parallel data. In order to adapt our Vega-MT to the general domains, we employ GREEDY BASED ENSEMBLING (§2.4.1), GENERALIZATION FINETUNING (§2.4.2) and POST-PROCESSING (§2.4.3) strategies.

| | $\mathcal{M}_{\textbf{Base}}$ | $\mathcal{M}_{\textbf{Big}}$ | $\mathcal{M}_{\textbf{XL}}$ |
|---|---|---|---|
| **#Stack** | 6 | 6 | 24 |
| **#Hidden_Size** | 512 | 1024 | 2048 |
| **#FFN_Size** | 2048 | 4096 | 16384 |
| **#Heads** | 8 | 16 | 32 |

Table 1: Model differences among base ( $\mathcal{M}_{\textbf{Base}}$ ), big ($\mathcal{M}_{\textbf{Big}}$) and extremely large ($\mathcal{M}_{\textbf{XL}}$).

The subsequent paper is designed as follows. We introduce the major approaches we used in Section 2. In Section 3, we provide the data description. We also present the experimental settings and results in Section 4. Conclusions are described in Section 5.

## 2 Approaches

### 2.1 Neural Machine Translation Frameworks

The neural machine translation task aims to transform a source language sentence into the target language with a neural network. There are several generation paradigms for translation, *e.g.* Autoregressive Translation (AT, Bahdanau et al., 2015b; Vaswani et al., 2017b) and Non-Autoregressive Translation (NAT, Gu et al., 2018).

**Autoregressive Translation** Given a source sentence $\mathbf{x}$, an NMT model generates each target word $\mathbf{y}_t$ conditioned on previously generated ones $\mathbf{y}_{<t}$. Accordingly, the probability of generating $\mathbf{y}$ is computed as:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}, \mathbf{y}_{<t}; \theta) \tag{1}$$

where $T$ is the length of the target sequence and the parameters $\theta$ are trained to maximize the likelihood of a set of training examples according to $\mathcal{L}(\theta) = \arg\max_{\theta} \log p(\mathbf{y}|\mathbf{x}; \theta)$. Typically, we choose Transformer (Vaswani et al., 2017b) as its state-of-the-art performance and scalability. We carefully employ the standard Transformer-BASE ($\mathcal{M}_{\textbf{Base}}$) and Transformer-BIG ($\mathcal{M}_{\textbf{Big}}$) in the preliminary studies, and also scale the framework up to an extremely large setting (Tran et al., 2021) – Transformer-XL ($\mathcal{M}_{\textbf{XL}}$) to maintain powerful

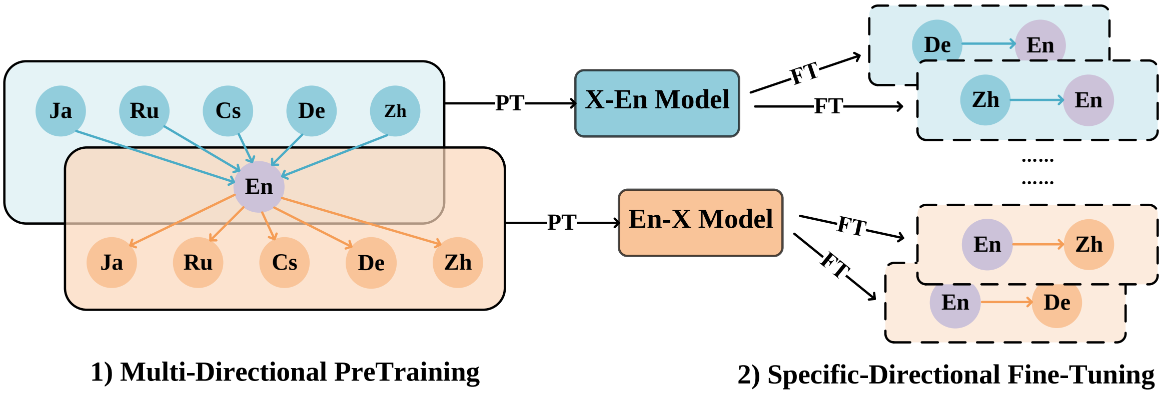


Figure 2: The schematic structure of the two main stages of the Vega-MT.

model capacity (see Table 1) . In Vega-MT, we utilized the autoregressive translation (AT) model with $\mathcal{M}_{\textbf{Big}}$ and $\mathcal{M}_{\textbf{XL}}$ for multi-directional pretraining (§2.2.1), specific-directional finetuning (§2.2.2), bidirectional self-training (§2.3.2) and generalization fine-tuning (§2.4.2) as its powerful modelling ability and generation accuracy.

**Non-Autoregressive Translation** Different to autoregressive translation (Bahdanau et al., 2015b; Vaswani et al., 2017b, AT) models that generate each target word conditioned on previously generated ones, non-autoregressive translation (Gu et al., 2018, NAT) models break the autoregressive factorization and produce the target words in parallel. Given a source sentence $\mathbf{x}$, the probability of generating its target sentence $\mathbf{y}$ with length $T$ is defined by NAT as:

$$p(\mathbf{y}|\mathbf{x}) = p_L(T|\mathbf{x};\theta)\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x};\theta) \tag{2}$$

where $p_L(\cdot)$ is a separate conditional distribution to predict the length of target sequence. Typicallly, most NAT models are implemented upon the framework of $\mathcal{M}_{\textbf{Base}}$. We utilized the NAT for bidirectional self-training (§2.3.2) as NAT can nicely avoid the error accumulation problems during generation, and generate diverse synthetic samples. Also, we employ several advanced structure (Gu et al., 2019; Ding et al., 2020) (*Levenshtein* with source local context modelling) and advanced training strategies (Ding et al., 2021a,b,c, 2022b; Ding, 2022) to obtain high quality and diverse translations.

## 2.2 Multidirectional Pretraining as Foundation Models

This section illustrates how we scale the "bidirectional" training in Ding et al. (2021d) up to "multi-directional" pretraining with all high-resource parallel corpora, including Zh, De, Cs, Ru, Ja to/from En. The pretrained foundation models will be finetuned for the downstream specific-directional task, *e.g.* Zh-En. Such two-stage scheme is shown in Figure 2.

### 2.2.1 Multi-Directional Pretraining

Recent works on real-world `WMT` translation datasets have verified that it is possible to transfer the pretrained cross-lingual knowledge to the downstream tasks with the pretrain-finetune paradigm, hence improving performance and generalization ability (Ding et al., 2022b,a; Wang et al., 2020a).

Here, we propose multi-directional pretraining by extending Bidirectional Pretraining (Ding et al., 2021d, BiT) to utilize multiple translation corpora of different languages. Compared with BiT, multi-directional pretraining could utilize the cross-lingual knowledge among more languages, thus further exploiting the cross-language knowledge and facilitating the downstream transferring. The main modifications could be summarized twofold:

1) We increase language numbers to utilize the cross-lingual knowledge of various languages. The straight setting for multi-directional pretraining is multi-lingual translation, which is divided into Many-to-Many (M2M), One-to-Many (O2M), and Many-to-One (M2O), according to the language number that the model supports. M2M has potential of capturing more cross-

lingual knowledge from $N * N$ pairs compared with $N * 1/1 * N$ pairs of M2O/O2M but usually leads to worse performance because of the imbalanced language data distribution question (Freitag and Firat, 2020). Inspired by (Tran et al., 2021), we focus on pretraining two separate systems, including English-to-Many and Many-to-English. We also prepend the corresponding language token to source & target sentences.

2) We further expand model size to an extremely large setting. While enjoying the benefit of cross-lingual knowledge transferring, the difficulty of modeling extremely large-scale data and language-specific feature pushes us to enlarge Transformer-BIG to an extremely large size (4.7 Billion parameters, see Table 1). This ensures our models are capable of better mastering multiple translation corpus.

#### 2.2.2 Specific-Directional Finetuning

The off-target problem, which widely exits in multilingual translation systems (Yang et al., 2021), indicates model often generates the translation with some non-target words. To reduce non-target word translation ratio in multi-directional pretrained models, we consider a two-stage specific-directional finetuning strategy. As shown in Figure 2, the English source/target model is tuned with an English source/target bilingual corpus.

Specifically, we first replace the multilingual embedding with a bilingual one. To fit model and bilingual vocabulary, we freeze all parameters of the Transformer backbone and only tune embedding layers in this stage. Next, we employ full model finetuning on large-scale translation corpus. This allows the model to fully adapt to the specific directional translation task, thus further achieving gains. To balance both finetune stages, we set the ratios of update step as $1 : 4$ for embedding- and full model-tuning, respectively.

For future work during specific directional finetuning, it will be interesting to design tuning data order (Liu et al., 2020a; Zhou et al., 2021) by leveraging the learning difficulty of each training sample estimated in the pretraining stage.

### 2.3 Data Augmentation Strategies

In Vega-MT, we consider augmenting both the parallel and monolingual data comprehensively. Specifically, we employ the cycle translation (Ding and Tao, 2019) for regenerating the low-quality *monolingual data*, and adopt bidirectional self-training (Ding and Tao, 2021) to distill, diversify *both the monolingual and parallel data*.

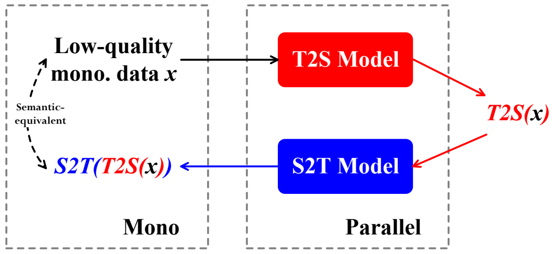


Figure 3: The Cycle Translation process, into which we feed the low quality monolingual data $x$, and then correspondingly obtain the improved data $\mathcal{CT}(x)$ (denoted as $S2T(T2S(x))$). Note that models marked in red and blue represent the target-to-source and source-to-target model trained with $\mathcal{M}_{\textbf{Big}}$. The dotted double-headed arrow between the input $\mathbf{x}$ and the final output $\mathcal{CT}(x)$ means they share the semantic but differ in fluency.

| # | Cycle Translated Sentence "1"→"2" |
|---|---|
| 1 | *She stuck to her principles even when some suggest that in an environment often considered devoid of such thing there are little point.* |
| 2 | *She insists on her own principles, even if some people think that it doesn't make sense in an environment that is often considered to be absent.* |

Table 2: Example of difference between original sentence (line 1) and cycle translated result (line 2). Pretrained BERT model using all available English corpora show that the $\mathcal{L}oss$ decreased from 6.98 to 1.52.

#### 2.3.1 Cycle Translation for Mono. Data

There is a large amount of monolingual data incomplete or grammatically incorrect. To fully leverage such part of monolingual data for better data augmentation, *e.g.* back translation (Sennrich et al., 2016) or sequence -level knowledge distillation (Kim and Rush, 2016), we adopt Cycle Translation (Ding and Tao, 2019) (denoted as $\mathcal{CT}(\cdot)$, as Figure 3) to improve the monolingual data below the quality-threshold (the latter 50% will be cycle translated according to Ding and Tao (2019)'s optimal setting). We give an example in Table 2 to clearly show how the cycle translation improves the quality of the sentence.

#### 2.3.2 Bidirectional Self-Training for Both Mono&Para Data

Currently, data-level methods have attracted the attention of the community, including exploiting the parallel and monolingual data. The most representative approaches include:

- Back Translation (**BT**, Sennrich et al. 2016) introduces the target-side monolingual data by translating with an inverse translation model, and combines the synthetic data with parallel data;

- Knowledge Distillation (**KD**, Kim and Rush 2016) generates the synthetic data with sequence-level knowledge distillation;

- Data Diversification (**DD**, Nguyen et al. 2020) diversifies the data by applying KD and BT on parallel data.

Clearly, self-training is at the core of above approaches, that is, they generate the synthetic data either from source to target or reversely, with either monolingual or bilingual data.

To this end, we employ the bidirectional self-training (Ding and Tao, 2021; Liao et al., 2020) strategy for both parallel and monolingual data (including source and target, respectively). Specifically, baseline AT models with $\mathcal{M}_{\textbf{Big}}$ setting and NAT models with $\mathcal{M}_{\textbf{Base}}$ setting are trained with original (distilled for NAT) parallel data in the first iteration, and based on these forward- and backward-teachers, all available source & target language sentences can be used to generate the corresponding synthetic target & source sentences. The authentic and synthetic data (generated by AT and NAT models) are then concatenated to train the second round AT and NAT models. We run the bidirectional self-training by totally 2 rounds for each translation direction. And for each round, we train 3 forward- and 3 backward- AT models, and 1 forward- and backward- NAT models to perform self-training. In this way, the amount of bidirectional synthetic data will be 8x larger than the original parallel and monolingual data.

### 2.4 Generalization Adaptation for Downstream Translation

To adapt Vega-MT to the general domain translation task, we employ several strategies, including ensembling, generalization finetuning, and post-processing. Note that in our preliminary study, we find that noisy channel reranking with the target-to-source MT model and language model does not work in our setting, thus we have not reranked the results in the final submission.

**Algorithm 1:** Generalization Finetuning with Iteratively Transductive Ensemble

```
Input: Single Model M_n,
       General Seed D={D_1, D_2..D_k},
       Ensemble N models E_N.
Output: New Model M'_n
t := 0
while not convergence do
    Translate D_1 with E_N and get D_1^{E_N}
    ..
    Translate D_k with E_N and get D_k^{E_N}
    D^{E_N} = D_1^{E_N} ∪ ..D_k^{E_N}
    Train M_n on D ∪ D^{E_N} and get M'_n,
      then M_n = M'_n
    t := t + 1
end
```

| | |
|---|---|
| SRC | *Siltalan edellinen kausi liigassa oli 2006-07* |
| HYP | *Siltala's previous season in the league was 2006 at 07* |
| +post | *Siltala's previous season in the league was 2006-07* |

Table 3: Example of the effectiveness of post-processing in handling inconsistent number translation.

#### 2.4.1 Greedy Based Ensembling

Greedy based ensembling adopts an easy operable greedy-base strategy to search for a better single model combinations on the development set, which consistently shows better performance than simply average in our preliminary study, therefore we technically follow the instruction of Deng et al. (2018) to choose the optimal combination of checkpoints to enhance the generalization and boost performance of the final model. We refer to this method as "Ensemble" in the following.

#### 2.4.2 Generalization Finetuning

As the general domain evaluation is on multi-domain directions, *i.e.* containing (up to) four dif-

| Languages | # Sents | # Ave. Len. |
|---|---|---|
| | *Parallel* | |
| ZH-EN | 46,590,547 | 22.8/27.1 |
| DE-EN | 292,020,383 | 22.9/21.7 |
| CS-EN | 88,244,832 | 20.5/19.9 |
| RU-EN | 98,454,430 | 28.5/27.8 |
| JA-EN | 28,943,024 | 26.2/28.0 |
| | *Monolingual* | |
| EN | 1,384,791,758 | 21.3 |
| ZH | 1,346,538,572 | 25.8 |
| DE | 5,612,161,001 | 23.2 |
| CS | 444,049,843 | 19.7 |
| RU | 8,351,860,471 | 28.5 |
| JA | 5,534,872,418 | 27.9 |

Table 4: Data statistics after pre-processing.

ferent domains, we design generalization finetuning strategy to transductively finetune (Wang et al., 2020b) on each domain, and ensemble them into one single model, to empower the general translation ability. The proposed generalization finetuning is shown in Algorithm 1. The main difference from Multi-Model & Multi-Iteration Transductive Ensemble (Wang et al., 2021) is that the $k_{th}$ domain seed $D_k$ is extracted from the test set using heuristic artificial knowledge.

### 2.4.3 Post-Processing

In addition to general post-processing strategies (*e.g.* de-BPE), we also employ a post-processing algorithm (Wang et al., 2018) for inconsistent number, date translation, for example, "*2006-07*" might be translated to the wrong translation "*2006 at 07*". Our post-processing algorithm will search for the best matching number string from the source sentence to replace these types of errors (see Table 3). Besides, we also conduct punctuation conversion, including convert quotation marks to German double-quote style (Czech, German), convert punctuation to language-specific characters (Japanese, Chinese).

## 3 Data Preparation

We participated in translation of all high-resource tracks and one medium-resource track, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En).

In this section, we take the En↔Zh translation as example and describe how to prepare the training data. The setting is the same for other language pairs. We use all available parallel corpus for En↔Zh [§], including ParaCrawl v9, News Commentary v16, Wiki Titles v3, UN Parallel Corpus V1.0, CCMT Corpus, WikiMatrix and Back-translated news. For monolingual data, we randomly sample from "News Crawl" and "Common Crawl". The final corpus statistics are presented in Table 4.

To improve the quality of parallel data, we further propose to filter the low-quality samples. First, we remove the sentence pair which is predicted as wrong language with `Fasttext` (Joulin et al., 2017, 2016). Second, we replace unicode punctuation and also normalize punctuation with `mosesdecoder`. We also remove duplicate sentence pairs and filter out sentences with illegal characters. For length, we remove sentences longer than 250 words and with a source/target length ratio exceeding 3.

## 4 Experiments

**Settings** We use the extremely large Transformer ($\mathcal{M}_{\mathbf{XL}}$) for all tasks and Transformer-BIG ($\mathcal{M}_{\mathbf{BIG}}$) for bilingual baselines. For $\mathcal{M}_{\mathbf{BIG}}$, we empirically adopt large batch strategy (Edunov et al., 2018) (*i.e.* 458K tokens/batch) to optimize the performance. The learning rate warms up to $1 \times 10^{-7}$ for 10K steps, and then decays for 70K steps with the cosine schedule. For regularization, we tune the dropout rate from [0.1, 0.2, 0.3] based on validation performance, and apply weight decay with 0.01 and label smoothing with $\epsilon$ = 0.1. We use Adam optimizer (Kingma and Ba, 2015) to train models. We evaluate the performance on an ensemble of last 10 checkpoints to avoid stochasticity. For the main model $\mathcal{M}_{\mathbf{XL}}$, we adopt 1M Tokens/Batch to optimize the performance both in multilingual pretraining and bilingual finetuning. We set 0.1 as the label smoothing ratio, 4000 as warm-up steps, and 1e-3 as the learning rate. We optimize Vega-MT with Adam (Kingma and Ba, 2015). We use 100k updates for multi-directional pretraining, 40k updates for each specific-directional finetuning. For

[§]both parallel and monolingual corpus can be obtained fromhttps://www.statmt.org/wmt22/translation-task.html

| | Zh-En | | | En-Zh | | |
|---|---|---|---|---|---|---|
| **Models** | **W21 test** | **W22 test** | $\Delta$ | **W21 test** | **W22 test** | $\Delta$ |
| **Transformer-BIG w/ Para.** | 25.3 | 21.9 | - | 25.9 | 33.2 | - |
| **Multi-Directional PT** | 28.4 | 25.1 | *+3.2* | 27.1 | 35.7 | *+1.9* |
| +Specific-Directional FT | 29.5 | 26.7 | *+4.3* | 27.4 | 36.6 | *+3.6* |
| +Bidirect. Self-Training | 30.8 | 29.0 | *+6.3* | 29.7 | 40.7 | *+5.7* |
| +Ensemble | **31.1** | 29.8 | *+6.7* | 30.4 | 41.3 | *+6.4* |
| +Generalization FT | 30.3 | **33.5** | *+8.3* | 30.6 | 44.1 | *+9.0* |
| +Post-Processing | 30.5 | **33.5** | ***+8.4*** | **33.6** | **49.7** | ***+13.3*** |

Table 5: **Ablation studies of each component on Zh↔En** general translation task in terms of SacreBLEU.We select Transformer-BIG only trained with official parallel data as the baseline.

| **Models** | **Zh→En** | **De→En** | **Cs→En** | **Ru→En** | **Ja→En** | $\Delta$ |
|---|---|---|---|---|---|---|
| **Baseline** | 21.9 | 23.0 | 42.5 | 30.2 | 19.0 | - |
| **Vega-MT** | **33.5** | **33.7** | **54.9** | 45.1 | 25.6 | ***+11.2*** |
| Best Official | 33.5 | 33.7 | 54.9 | **45.1** | **26.6** | |
| **Models** | **En→Zh** | **En→De** | **En→Cs** | **En→Ru** | **En→Ja** | $\Delta$ |
| **Baseline** | 33.2 | 26.4 | 34.8 | 20.8 | 17.9 | - |
| **Vega-MT** | **49.7** | **37.8** | **41.4** | **32.7** | 41.5 | ***+14.0*** |
| Best Official | 49.7 | 37.8 | 41.4 | 32.7 | **42.5** | |

Table 6: **SacreBLEU-Scores of our submissions in WMT2022 general translation task.** "Baseline" indicates the performance of the baseline systems. And "Best Official" denotes the best results of constrained systems in each direction.

evaluation, we select SacreBLEU (Post, 2018) as the metric for all tasks. `news-test2020` and `news-test2021` are selected for validation and test respectively.

All parallel data will be used in the multi-directional PT stage, and during specific-directional FT, corresponding bilingual data augmented by bidirectional self-training are utilized. Each sentence are jointly tokenized in to sub-word units with SentencePiece (Kudo and Richardson, 2018), which is trained on all concatenated multilingual parallel data for Transformer-XL with merge operation 120K at the pretraining stage, and during finetuning stage, is trained on corresponding bilingual data with merge operation 60K for English and 75K for other languages. And for each baseline with Transformer-BIG, the joint bilingual vocab size is 80K. During pretraining, we select the sample with temperature-based method (T=5) to preserve the representation of relatively low-resource language, *e.g.* Japanese. We grid-search the beam size within the range of [3,4,5,..,8] on validation set for each translation task. All models are trained on 32 DGX-SuperPOD A100 GPUs for about two weeks pre-training and five days fine-tuning.

**Main Results** To illustrate the effectiveness of each strategy in our Vega-MT, we report the ablation results in Table 5 on Zh↔En tasks. Clearly, directly generating the translations with the multi-directional pretrained model could obtain average +3.2 and +1.9 BLEU improvements for Zh-En and En-Zh, respectively, which is consistent with the findings of Tran et al. (2021). We show that tuning on downstream bilingual data could further improve the translation by +1.4 BLEU points, showing the necessity of bridging the cross-lingual gap with in-domain learning during leveraging multilingual pretrain (Zan et al., 2022a). Bidirectional self-training actually contains several strategies, *e.g.* back translation, distillation and data diversification, and we empirically show that such data augmentation strategy nicely complement pretraining, which is also verified by Liu et al. (2021). Other strategies could consistently enhance the translation performance besides the generalization FT for the news domain

| Models | Zh→En | De→En | Cs→En | Ru→En | Ja→En | Δ |
|---|---|---|---|---|---|---|
| **Baseline** | 16.5 | 3.5 | 40.1 | 8.5 | 21.5 | - |
| **Vega-MT** | **45.1** | **58.0** | **74.7** | **64.9** | 40.6 | ***+38.6*** |
| Best Official | 45.1 | 58.0 | 74.7 | 64.9 | **42.0** | |
| **Models** | **En→Zh** | **En→De** | **En→Cs** | **En→Ru** | **En→Ja** | **Δ** |
| **Baseline** | 26.6 | -40.6 | 66.9 | -1.4 | 42.1 | - |
| **Vega-MT** | **61.7** | **63.2** | 95.3 | **69.6** | **65.1** | ***+52.3*** |
| Best Official | 61.7 | 63.2 | **96.0** | 69.6 | 65.1 | |

Table 7: **COMET-Scores of our submissions in WMT2022 general translation task.** "Baseline" indicates the performance of the baseline systems. And "Best Official" denotes the best results of constrained systems in each direction.

test2021, where the Zh-En model decreases the BLEU scores (-0.8 BLEU) because the generalization FT is designed and tuned for the general domain test2022.

Table 6 and Table 7 show the final submissions in terms of SacreBLEU and COMET scores, including Zh, De, Cs, Ru and Ja to/from En, listing the baseline and our final submissions. We also report the best official scores among all constrained systems "Best Official" as reference. As seen, SacreBLEU and COMET results show identical trends, where our Vega-MT outperforms baseline Transformer-BIG by +11.2/ +38.6 and +14.0/ +52.3 BLEU/ COMET points, showing the effectiveness and universality of our model. Interestingly, we observe that the improvements upon En-X are more significant than that of X-En, which will be investigated in our future work. For more system rankings, please refer Table 8 and Table 9 in Appendix for SacreBLEU and COMET results, respectively.

## 5 Conclusion

This paper presents the JD Explore Academy machine translation system Vega-MT for WMT 2022 shared tasks on general machine translation. We investigate various frameworks, including autoregressive and non-autoregressive Transformer with BASE, BIG and XL settings, respectively, to build strong baseline models. Then we push the limit of bidirectional training by scaling up two main factors, *i.e.* language pairs and model scales, to develop the powerful foundation Vega-MT model. Also, the popular data augmentation methods, *e.g.* cycle translation and bidirectional self-training, are combined to improve their performance. We carefully design the generalization adaptation strategies to further improve the multi-domain performance. Among all participated constrained systems, our Vega-MT won 7 champions, 2 runners-up and 1 third place w.r.t sacreBLEU. And according to the COMET, we won 8 champions and 2 runners-up.


## Acknowledgments

The authors wish to thank the organizers of WMT2022 for their great efforts in the organization, and their prompt responses to our questions. The authors are grateful to the anonymous reviewers for their insightful comments and careful proofreading. The authors also specially thank Yukang Zhang (JD Explore Academy), who kindly supports us by maintaining a stable computing platform.


## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015a. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015b. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for WMT18. In *WMT*.

Liang Ding. 2022. *Neural Machine Translation with Fully Information Transformation*. Ph.D. thesis, The University of Sydney.

Liang Ding, Keqin Peng, and Dacheng Tao. 2022a. Improving neural machine translation by denoising training. *arXiv preprint*.

Liang Ding and Dacheng Tao. 2019. The University of Sydney's machine translation system for WMT19. In *WMT*.

Liang Ding and Dacheng Tao. 2021. The USYD-JD speech translation system for IWSLT2021. In *IWSLT*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021a. Progressive multi-granularity training for non-autoregressive translation. In *findings of ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021c. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Liang Ding, Longyue Wang, Shuming Shi, Dacheng Tao, and Zhaopeng Tu. 2022b. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020. Context-aware cross-attention for non-autoregressive translation. In *COLING*.

Liang Ding, Di Wu, and Dacheng Tao. 2021d. Improving neural machine translation by bidirectional training. In *EMNLP*.

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

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *WMT*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *NeurIPS*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Baohao Liao, Yingbo Gao, and Hermann Ney. 2020. Multi-agent mutual learning at sentence-level and token-level for neural machine translation. In *Findings of EMNLP 2020*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *EMNLP*.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020a. Norm-based curriculum learning for neural machine translation. In *ACL*.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, Shuming Shi, and Zhaopeng Tu. 2021. On the complementarity between pre-training and back-translation for neural machine translation. In *EMNLP*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual denoising pre-training for neural machine translation. *TACL*.

Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. In *NeurIPS*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *ICML*.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *NeurIPS*.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the WMT21 news translation task. In *WMT*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI lab machine translation systems for WMT20 chat translation task. In *WMT*.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The NiuTrans machine translation system for WMT18. In *WMT*.

Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. 2022. Understanding and improving sequence-to-sequence pretraining for neural machine translation. In *ACL*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020b. Transductive ensemble learning for neural machine translation. In *AAAI*.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *EMNLP*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022a. Bridging cross-lingual gaps during leveraging the multilingual sequence-to-sequence pretraining for text generation. *arXiv preprint*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022b. On the complementarity between pre-training and random-initialization for resource-rich machine translation. In *COLING*.

Lei Zhou, Liang Ding, Kevin Duh, Shinji Watanabe, Ryohei Sasano, and Koichi Takeda. 2021. Self-guided curriculum learning for neural machine translation. In *IWSLT*.

| pair | system | id | is_constrained | metric | score |
|---|---|---|---|---|---|
| **En-Cs** | Lan-Bridge | 551 | FALSE | bleu-B | 45.6 |
| **En-Cs** | JDExploreAcademy | 829 | TRUE | bleu-B | **41.4** |
| **En-Cs** | CUNI-DocTransformer | 800 | TRUE | bleu-B | 39.8 |
| **En-Cs** | CUNI-Bergamot | 734 | TRUE | bleu-B | 38.6 |
| **En-Cs** | CUNI-Transformer | 761 | TRUE | bleu-B | 37.7 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-De** | JDExploreAcademy | 843 | TRUE | bleu-A | **37.8** |
| **En-De** | Lan-Bridge | 549 | FALSE | bleu-A | 36.1 |
| **En-De** | PROMT | 694 | FALSE | bleu-A | 36.1 |
| **En-De** | OpenNMT | 207 | FALSE | bleu-A | 35.7 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Ja** | NT5 | 763 | TRUE | bleu-A | 42.5 |
| **En-Ja** | DLUT | 789 | TRUE | bleu-A | 41.8 |
| **En-Ja** | LanguageX | 676 | FALSE | bleu-A | 41.7 |
| **En-Ja** | JDExploreAcademy | 516 | TRUE | bleu-A | **41.5** |
| **En-Ja** | Lan-Bridge | 555 | FALSE | bleu-A | 39.4 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Ru** | JDExploreAcademy | 509 | TRUE | bleu-A | **32.7** |
| **En-Ru** | Lan-Bridge | 556 | FALSE | bleu-A | 32.6 |
| **En-Ru** | HuaweiTSC | 680 | TRUE | bleu-A | 30.8 |
| **En-Ru** | PROMT | 804 | FALSE | bleu-A | 30.6 |
| **En-Ru** | SRPOL | 265 | TRUE | bleu-A | 30.4 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Zh** | LanguageX | 716 | FALSE | bleu-A | 54.3 |
| **En-Zh** | HuaweiTSC | 557 | FALSE | bleu-A | 49.7 |
| **En-Zh** | JDExploreAcademy | 834 | TRUE | bleu-A | **49.7** |
| **En-Zh** | AISP-SJTU | 611 | TRUE | bleu-A | 48.8 |
| **En-Zh** | Manifold | 336 | TRUE | bleu-A | 48.7 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Cs-En** | JDExploreAcademy | 505 | TRUE | bleu-B | **54.9** |
| **Cs-En** | Lan-Bridge | 585 | FALSE | bleu-B | 54.5 |
| **Cs-En** | CUNI-DocTransformer | 805 | TRUE | bleu-B | 51.9 |
| **Cs-En** | CUNI-Transformer | 772 | TRUE | bleu-B | 51.6 |
| **Cs-En** | SHOPLINE-PL | 819 | TRUE | bleu-B | 46.8 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **De-En** | JDExploreAcademy | 809 | TRUE | bleu-A | **33.7** |
| **De-En** | Lan-Bridge | 587 | FALSE | bleu-A | 33.4 |
| **De-En** | PROMT | 796 | FALSE | bleu-A | 32.5 |
| **De-En** | LT22 | 605 | TRUE | bleu-A | 26.0 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Ja-En** | NT5 | 766 | TRUE | bleu-A | 26.6 |
| **Ja-En** | JDExploreAcademy | 512 | TRUE | bleu-A | **25.6** |
| **Ja-En** | DLUT | 693 | TRUE | bleu-A | 24.8 |
| **Ja-En** | Lan-Bridge | 588 | FALSE | bleu-A | 22.8 |
| **Ja-En** | NAIST-NICT-TIT | 583 | TRUE | bleu-A | 22.7 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Ru-En** | Lan-Bridge | 589 | FALSE | bleu-A | 45.2 |
| **Ru-En** | HuaweiTSC | 836 | TRUE | bleu-A | 45.1 |
| **Ru-En** | JDExploreAcademy | 769 | TRUE | bleu-A | **45.1** |
| **Ru-En** | SRPOL | 666 | TRUE | bleu-A | 43.6 |
| **Ru-En** | ALMAnaCH-Inria | 710 | TRUE | bleu-A | 30.3 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Zh-En** | JDExploreAcademy | 708 | TRUE | bleu-A | **33.5** |
| **Zh-En** | LanguageX | 219 | FALSE | bleu-A | 31.9 |
| **Zh-En** | HuaweiTSC | 477 | FALSE | bleu-A | 29.8 |
| **Zh-En** | AISP-SJTU | 648 | TRUE | bleu-A | 29.7 |
| **Zh-En** | Lan-Bridge | 386 | FALSE | bleu-A | 28.1 |



Table 8: **Ranking of our submissions in terms of SacreBLEU-Score** in WMT2022 general translation task.

| pair | system | id | is_constrained | metric | score |
|---|---|---|---|---|---|
| **En-Cs** | CUNI-Bergamot | 734 | TRUE | COMET-B | 0.960 |
| **En-Cs** | JDExploreAcademy | 829 | TRUE | COMET-B | **0.953** |
| **En-Cs** | Lan-Bridge | 551 | FALSE | COMET-B | 0.947 |
| **En-Cs** | CUNI-DocTransformer | 800 | TRUE | COMET-B | 0.917 |
| **En-Cs** | CUNI-Transformer | 761 | TRUE | COMET-B | 0.866 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-De** | JDExploreAcademy | 843 | TRUE | COMET-A | **0.632** |
| **En-De** | Lan-Bridge | 549 | FALSE | COMET-A | 0.588 |
| **En-De** | OpenNMT | 207 | FALSE | COMET-A | 0.572 |
| **En-De** | PROMT | 694 | FALSE | COMET-A | 0.558 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Ja** | JDExploreAcademy | 516 | TRUE | COMET-A | **0.651** |
| **En-Ja** | NT5 | 763 | TRUE | COMET-A | 0.641 |
| **En-Ja** | LanguageX | 676 | FALSE | COMET-A | 0.621 |
| **En-Ja** | DLUT | 789 | TRUE | COMET-A | 0.605 |
| **En-Ja** | Lan-Bridge | 555 | FALSE | COMET-A | 0.565 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Ru** | JDExploreAcademy | 509 | TRUE | COMET-A | **0.696** |
| **En-Ru** | Lan-Bridge | 556 | FALSE | COMET-A | 0.673 |
| **En-Ru** | PROMT | 804 | FALSE | COMET-A | 0.603 |
| **En-Ru** | SRPOL | 265 | TRUE | COMET-A | 0.597 |
| **En-Ru** | HuaweiTSC | 680 | TRUE | COMET-A | 0.592 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **En-Zh** | LanguageX | 716 | FALSE | COMET-A | 0.638 |
| **En-Zh** | JDExploreAcademy | 834 | TRUE | COMET-A | **0.617** |
| **En-Zh** | Lan-Bridge | 714 | FALSE | COMET-A | 0.614 |
| **En-Zh** | Manifold | 336 | TRUE | COMET-A | 0.601 |
| **En-Zh** | HuaweiTSC | 557 | FALSE | COMET-A | 0.595 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Cs-En** | JDExploreAcademy | 505 | TRUE | COMET-B | **0.747** |
| **Cs-En** | Lan-Bridge | 585 | FALSE | COMET-B | 0.718 |
| **Cs-En** | CUNI-DocTransformer | 805 | TRUE | COMET-B | 0.706 |
| **Cs-En** | CUNI-Transformer | 772 | TRUE | COMET-B | 0.692 |
| **Cs-En** | SHOPLINE-PL | 819 | TRUE | COMET-B | 0.611 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **De-En** | JDExploreAcademy | 809 | TRUE | COMET-A | **0.580** |
| **De-En** | Lan-Bridge | 587 | FALSE | COMET-A | 0.565 |
| **De-En** | PROMT | 796 | FALSE | COMET-A | 0.518 |
| **De-En** | LT22 | 605 | TRUE | COMET-A | 0.256 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Ja-En** | NT5 | 766 | TRUE | COMET-A | 0.420 |
| **Ja-En** | JDExploreAcademy | 512 | TRUE | COMET-A | **0.406** |
| **Ja-En** | DLUT | 693 | TRUE | COMET-A | 0.372 |
| **Ja-En** | NAIST-NICT-TIT | 583 | TRUE | COMET-A | 0.334 |
| **Ja-En** | LanguageX | 435 | FALSE | COMET-A | 0.329 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Ru-En** | JDExploreAcademy | 769 | TRUE | COMET-A | **0.649** |
| **Ru-En** | Lan-Bridge | 589 | FALSE | COMET-A | 0.631 |
| **Ru-En** | HuaweiTSC | 836 | TRUE | COMET-A | 0.609 |
| **Ru-En** | SRPOL | 666 | TRUE | COMET-A | 0.595 |
| **Ru-En** | ALMAnaCH-Inria | 710 | TRUE | COMET-A | 0.268 |
| **pair** | **system** | **id** | **is_constrained** | **metric** | **score** |
| **Zh-En** | JDExploreAcademy | 708 | TRUE | COMET-A | **0.451** |
| **Zh-En** | LanguageX | 219 | FALSE | COMET-A | 0.449 |
| **Zh-En** | Lan-Bridge | 386 | FALSE | COMET-A | 0.430 |
| **Zh-En** | HuaweiTSC | 477 | FALSE | COMET-A | 0.428 |
| **Zh-En** | AISP-SJTU | 648 | TRUE | COMET-A | 0.416 |



Table 9: **Ranking of our submissions in terms of COMET-Score** in WMT2022 general translation task.