Guiding Neural Machine Translation with Semantic Kernels

Ping Guo[1,2], Yue Hu[1,2,*], Xiangpeng Wei[3], Yubing Ren[1,2], Yunpeng Li[1,2], Luxi Xing[4], Yuqiang Xie[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Alibaba DAMO Academy, Hangzhou, China [4]Alibaba Group, Beijing, China
[1,2]{guoping,huyue,renyubing,liyunpeng,xieyuqiang}@iie.ac.cn
[3]pemywei@gmail.com [4]xingluxixlx@gmail.com

Abstract

Machine Translation task has made great progress with the help of auto-regressive decoding paradigm and Transformer architecture. In this paradigm, though the encoder can obtain global source representations, the decoder can only use translation history to determine the current word. Previous promising works attempted to address this issue by applying a draft or a fixed-length semantic embedding as target-side global information. However, these methods either degrade model efficiency or show limitations in expressing semantics. Motivated by Functional Equivalence Theory, we extract several **semantic kernels** from a source sentence, each of which can express one semantic segment of the original sentence. Together, these semantic kernels can capture global semantic information, and we project them into target embedding space to guide target sentence generation. We further force our model to use semantic kernels at each decoding step through an **adaptive mask** algorithm. Empirical studies on various machine translation benchmarks show that our approach gains approximately an improvement of 1 BLEU score on most benchmarks over the Transformer baseline and about 1.7 times faster than previous works on average at inference time.

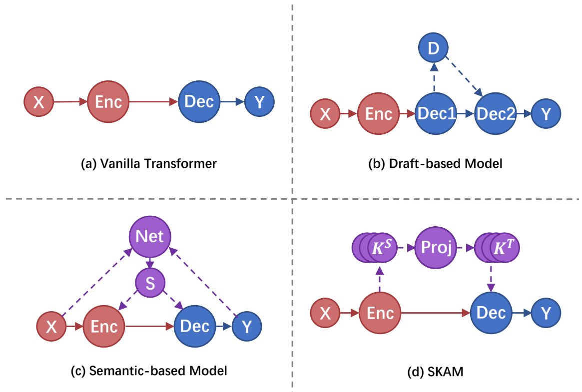

Figure 1: Comparison among methods with target-side global information. “red”, “blue” and “purple” color indicate source space, target space and semantic space, respectively. In (b), “D” means the draft generated by the first decoder. In (c), “Net” denotes the inference model in Semantic-based model and “S” is the semantic embedding. (d) shows our SKAM model, where “$\mathcal{K}^S$” and “$\mathcal{K}^T$” represent source and target semantic kernels, respectively. “Proj” is our projector.

1 Introduction

Machine Translation has been a long-standing task in natural language processing (Brown et al., 1990). Recently, Neural-based Machine Translation (NMT) models (Bahdanau et al., 2015; Wu et al., 2016; Vaswani et al., 2017) have made great progress and become the mainstream of machine translation frameworks. Most NMT models adopt the encoder-decoder framework. The encoder transforms the source sentence into source-side global representations. And the decoder generates the target sentence auto-regressively, based on the source-side representations and translation history.

However, one limitation of such auto-regressive decoding is that the generation of word y_t only has access to target-side partial information $y_{<t}$. If translation history is mistranslated, this error will be propagated to all subsequent words (Bengio et al., 2015). Also, this makes the generation heavily dependent on the source sentence, and minor changes in source sentence may lead to dramatic degradation in translation outcome (Cheng et al., 2019). Intuitively, using target-side global information to guide translation progress can alleviate this problem.

Attempts have been made to apply global information to guide the decoding process. Basically, we categorize them into two main lines. One is *draft* (Xia et al., 2017; Wang et al., 2019; Li et al., 2018; Zhang et al., 2018; Zhou et al., 2019), which generates a coarse target sequence to guide the translation progress, as depicted in Figure 1 (b). However, a coarse draft sentence requires delicate

*Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 7345 - 7356
December 7-11, 2022 ©2022 Association for Computational Linguistics

design to be generated. Thus, these methods often require multiple decoding steps. The other one is *latent semantics* (Shah and Barber, 2018; Zheng et al., 2020; Ai and Fang, 2021; Eikema and Aziz, 2019; Zhang et al., 2016; Su et al., 2018), which adopts generative methods (i.e., VAE (Kingma and Welling, 2014)) to model the semantics of source and target sentences in the latent semantic space. As in Figure 1 (c), such methods usually project semantics into one fixed-length vector, which shows limitations in expressing semantics for long sentences. Although above methods have successfully injected global information into decoding progress, they both incur extra computational cost, which greatly degrades the inference time compared to vanilla transformer model.

Motivated by the Functional Equivalence Theory (Nida and Taber, 1982), we propose Semantic Kernels with Adaptive Mask (SKAM) for NMT. To guide translation, we extract several semantic kernels from source sentence, each of which can express one semantic segment of the original sentence, as shown in Figure 1 (d). All semantic kernels together can capture the essential meaning of the source sentence, and they are later mapped from source space to target space with N-gram smoothing loss as target-side global information. We also improve auto-regressive decoding with an adaptive mask mechanism to guarantee the usage of semantic kernels in decoding progress. We evaluate the performance on several MT benchmarks that cover various data scales, languages and domains. Experiments show that our approach achieves significant improvement compared to the baselines and is about 1.7 times faster at inference than previous works on average. In total, our contributions can be summarized as:

- Inspired by Functional Equivalence Theory, we extract several semantic kernels from a source sentence to capture source semantics, which express sentence semantics at a new granularity.
- To map semantic kernels from source-side to target-side, we propose an N-gram smoothing loss, which guarantees each semantic kernel to capture one semantic segment, not one specific word.
- We design an adaptive mask mechanism to guarantee each decoding step can access comprehensive information, both preceding words (translation history) and subsequent words (semantic kernels).

2 Preliminaries and Related Work

2.1 Functional Equivalence Theory

The main point of Functional Equivalence Theory (Nida and Taber, 1982) is that translation should focus on the functional equivalence of information (sense-for-sense translation) rather than the direct formal equivalence (word-for-word translation). To do this, Nida and Taber (1982) proposes a translation framework, which consists of three parts:

Decompose: To get rid of the complex and ambiguous structure of the source sentence, the source sentence is split into several simple, short sentences, each of which captures one semantic segment of the original sentence. These simple sentences are called "kernel sentences", based on Transformational Generative Grammar (Chomsky, 2009).

Transfer: The kernel sentences are translated into receptor language. For the simplicity of the kernel sentences, they can be translated easily. And the translated kernel sentences can capture all source semantics, since languages agree far more on the level of the kernel sentences than on the level of the more elaborate structures (Nida and Taber, 1982).

Restructure: Transferred kernel sentences are restructured semantically and stylistically into the surface structure of target language.

Inspired by this theory, we try to make the translation comply more with source sentence meanings 'than source words in NMT model. Hence, we propose SKAM, which first decomposes source sentence to form semantic kernels (Kernel Selection Module), then transfers the semantic kernels into target embedding space (Kernel Projection Module), and finally restructures to a target sentence (Decoding Module).

2.2 Neural Machine Translation

Formally, let $X = \{x_0, x_1, ..., x_I\}$ and $Y = \{y_0, y_1, ..., y_J\}$ denote a source and a target sequence respectively, where I and J are the sentence lengths. Given a bilingual sentence pair $\langle X, Y \rangle$, an NMT model learns a set of parameters Θ to maximize the posterior probability $P(Y|X;\Theta)$:

$$P(Y|X;\Theta) = \prod_{t=0}^{J} P(y_t|y_{<t}, X;\Theta) \qquad (1)$$

where $y_{<t}$ is the partial translation that contains the target tokens before position t.

2.3 Transformer

Transformer model is based solely on attention mechanism. Given query Q, key K and value V, the output $\textsc{Att}(\mathrm{Q}, \mathrm{K}, \mathrm{V})$ is calculated as:

$$\textsc{Att}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}(\frac{\mathrm{QK}^\top}{\sqrt{d}})\mathrm{V} \qquad (2)$$

where $\sqrt{d}$ is the scaling factor with d being the dimension of embedding size.

Transformer model employs multiple-layer encoder and decoder to perform the translation task with residual connections among layers. Denote the output of the l-th layer as H^l, the encoder calculates:

$$\begin{aligned} \mathrm{O}_e^l &= \textsc{Att}(\mathrm{H}_e^{l-1}, \mathrm{H}_e^{l-1}, \mathrm{H}_e^{l-1}) + \mathrm{H}_e^{l-1} \\ \mathrm{H}_e^l &= \textsc{Ln}(\textsc{Ffn}(\textsc{Ln}(\mathrm{O}_e^l)) + \textsc{Ln}(\mathrm{O}_e^l)) \end{aligned} \qquad (3)$$

where $\textsc{Ln}(\cdot)$ and $\textsc{Ffn}(\cdot)$ are layer normalization and feed-forward networks with ReLU activation in between. As all of the Q, K, V come from the same place, this attention is referred to as self-attention.

The decoder is similar in structure to the encoder except that it includes another attention mechanism, called cross-attention, which attends to the output of the encoder stack H_e^L:

$$\begin{aligned} \mathrm{O}_d^l &= \textsc{Att}(\mathrm{H}_d^{l-1}, \mathrm{H}_d^{l-1}, \mathrm{H}_d^{l-1}) + \mathrm{H}_d^{l-1} \\ \mathrm{S}_d^l &= \textsc{Att}(\textsc{Ln}(\mathrm{O}_d^l), \mathrm{H}_e^L, \mathrm{H}_e^L) + \textsc{Ln}(\mathrm{O}_d^l) \\ \mathrm{H}_d^l &= \textsc{Ln}(\textsc{Ffn}(\textsc{Ln}(\mathrm{S}_d^l)) + \textsc{Ln}(\mathrm{S}_d^l)) \end{aligned} \qquad (4)$$

where the top layer of the decoder H_d^L is used to generate the final output sequence.

2.4 Target Information Enhanced NMT

Some impressive works have considered adding target information for better translation quality. Most closely related to our work are Deliberation Network (Xia et al., 2017) and Soft-prototype (Wang et al., 2019). These methods first generate a coarse draft to guide translation progress. Their main idea is to deliberate the wrong parts in the previous decoding step. Some other works have adopted bidirectional decoding (Li et al., 2018; Zhang et al., 2018; Zhou et al., 2019) or multi-pass decoding (Geng et al., 2018). Ma et al. (2018) applies target bag of words as targets to train NMT model. In comparison, our motivation is to extract semantic kernels that capture the essential meanings of the source sentence, and replenish these semantic segments to form a final target sentence.

Also related are the works of Zheng et al. (2020); Ai and Fang (2021); Shah and Barber (2018); Zhang et al. (2016); Su et al. (2018), which apply generative methods (VAE (Kingma and Welling, 2014)) to sample latent semantic embedding. Compared with these methods, we select different numbers of semantic kernels according to source sentence and avoid the EM-like decoding progress, which is more expressive and efficient.

In work similar to SKAM, Zhao et al. (2018) and Wang et al. (2017) integrate a phrase memory from a phrase-based statistical machine translation (SMT) system to guide the NMT model. Niehues et al. (2016) first adopts a phrase-based SMT system to pre-translate and then generates the final translation with an NMT model. However, these methods can not work without an SMT system at inference time, which limits their usage for translation.

3 NMT with Semantic Kernels

To make NMT model comply more with source sentence meaning than source sentence form, we propose SKAM, which consists of three modules: kernel selection module, kernel projection module, and decoding module, as depicted in Figure 2. We will explain each module in the following section.

3.1 Semantic Kernels Selection

Semantic kernels aim at capturing the essential meaning of the source sentence, and each of them should contain a semantic segment of the original sentence. Following Nida and Taber (1982), which claims that words acquire meaning through their context, we apply the contextual embedding of the content words to represent semantic kernels. Formally, semantic kernels are defined as:

$$\mathcal{K}^S(X) := \{\textsc{Enc}(x_i|X) \mid s(x_i) > 0,\ x_i \in X\} \qquad (5)$$

where $\textsc{Enc}$ denotes transformer encoder and $s(\cdot)$ is a norm-based significance score to locate the content words of the source sentence. To be mentioned, this definition of semantic kernel is simple, we will try to extract semantic kernels directly from the latent semantic space in future works.

Norm-based Significance Score

The significance score measures the ability of words to express essential meaning using the L2-

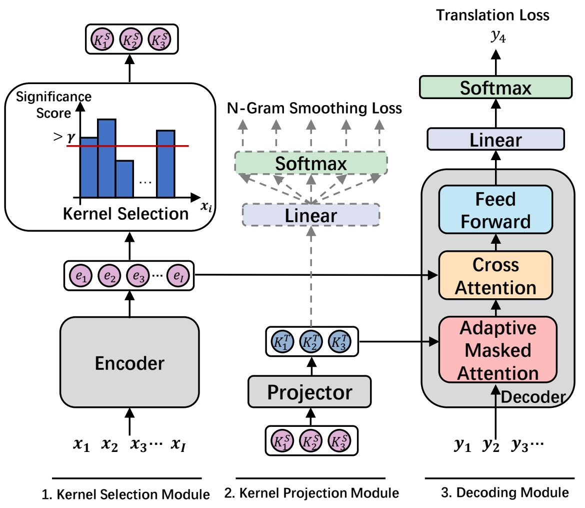

Figure 2: An overview of our SKAM model. SKAM contains three modules: 1. kernel selection module, to extract semantic kernels from source sentence; 2. kernel projection module, to map semantic kernels into target latent space; and 3. decoding module, which receives comprehensive target information via adaptive attention module. The N-gram Smoothng loss (dashed block) is only applied during training process.

norm of the word embedding. Intuitively, words that have higher L2-norms will play a leading role when adding up all word embeddings to form a sentence embedding. This feature of L2-norm has already been proven by some promising previous works (Luhn, 1958; Chen et al., 2020a; Liu et al., 2020).

We use the embedding matrix in our model to calculate L2-norm. As the norm of embedding matrix varies during training process, we scale each word norm $||x_i||$ with the current largest word norm $\max_{v \in V_S}(||v||)$ in source embedding. Our significance score $s(\cdot)$ is formulated as:

$$s(x_i) = \frac{||x_i||}{\max_{v \in V_S}(||v||)} - \gamma \quad (6)$$

where $\gamma \in [0, 1]$ is a norm threshold value. We only choose words whose score $s(x_i)$ is larger than γ as content words. To better understand what kinds of words are selected by Norm-based Significance Score, we sample some cases and illustrate them in Appendix A.

3.2 Semantic Kernels Projection

We try to apply a projector to map source-side semantic kernels $\mathcal{K}^S$ to target-side $\mathcal{K}^T$:

$$\mathcal{K}^T = f_{S \to T}(\mathcal{K}^S) \quad (7)$$

where $f_{S \to T}$ is a neural projector, $\mathcal{K}^S, \mathcal{K}^T \in \mathbb{R}^{Q \times d}$, Q is the number of semantic kernels and d means embedding size.

For words acquire meaning through their context (Nida and Taber, 1982), we train the projector to predict both content words and their context to better capture the deep meaning beneath surface expression. We propose N-gram smoothing loss to train the projector to concentrate on representing meaning, not a specific word.

N-gram Smoothing Loss

Given the encoder output of each source word $\textsc{Enc}(x_i|X)$, the Projector is trained to predict the corresponding target N-gram span $\text{Span}(y_i)$. We apply external alignment tool to find the aligned target word $\tilde{y}_i$ and group every N consecutive target words as an N-gram span. Formally,

$$\text{Span}(y_i) = \{\tilde{y}_{i-k}, \tilde{y}_{i-k+1}, ..., \tilde{y}_i, ..., \tilde{y}_{i+k-1}, \tilde{y}_{i+k}\} \quad (8)$$

where $k = (N-1)/2$ and N is a hyper-parameter to control how many words we select each time. The N-gram span is then used as label to train the projector with $\textsc{Enc}(x_i|X)$ as input. The N-gram smoothing loss $\mathcal{L}_g$ for one sample X formulates:

$$\mathcal{L}_g = \sum_{x_i \in X} \frac{1}{N} \sum_{m=0}^{2k} \log P(\tilde{y}_{i-k+m}|\textsc{Enc}(x_i|X)) \quad (9)$$

The output word embedding matrix in projector shares the same parameters with decoder and it is removed at inference time, as shown in Figure 2.

3.3 Decoding with Semantic Kernels

To give decoding progress comprehensive target-side information, we modify the original self-attention module in decoder to adaptive attention module, which can utilize both preceding words (from translation history) and subsequent words (from semantic kernels) to predict. Specifically, we concatenate semantic kernels to the K, V parts of the self-attention module in all decoder layers.

$$\textsc{Att}(\text{H}_d^{l-1}, [\mathcal{K}^T : \text{H}_d^{l-1}], [\mathcal{K}^T : \text{H}_d^{l-1}]) \quad (10)$$

Similar to Zheng et al. (2019), we explicitly separate semantic kernels into two groups: fully-accessed and not-yet-accessed. As translation progresses, we propose an Adaptive Mask to gradually remove the semantic kernels fully-accessed in translation history.

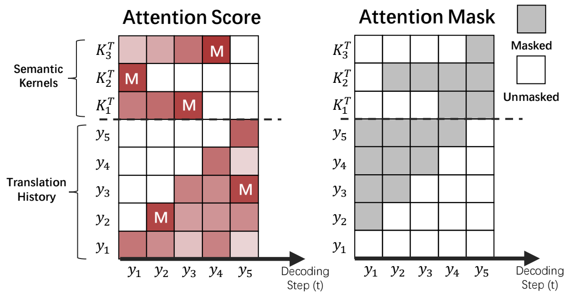

Figure 3: An illustration of our adaptive mask mechanism. The white "M" indicates the maximum attention score at current time step. After one semantic kernel gets the highest attention score, we mask it in the subsequent decoding step.

Adaptive Mask

Assuming 0 means unmask operation and 1 indicates mask operation, the attention mask $\mathcal{M}$ for semantic kernels should be like:

$$\mathcal{M}(\kappa_q^T, y_t) = \begin{cases} 0, & \kappa_q^T \text{ is not contained in } y_{<t} \\ 1, & \kappa_q^T \text{ is contained in } y_{<t} \end{cases} \tag{11}$$

where $\kappa_q^T \in \mathcal{K}^T$ and q means the q-th semantic kernel. We use the previous attention score $A_{<t}$ as a measurement whether semantic kernel κ_q^T has been fully-accessed in translation history. That is to say, if κ_q^T appears to have the largest attention score at time step t, we assume κ_q^T is fully-accessed at time step t and mask it in subsequent time steps, as illustrated in Figure 3. Formally, we update attention mask $\mathcal{M}(\kappa_q^T, y_t)$ based on previous attention mask and attention score:

$$\mathcal{M}(\kappa_q^T, y_t) = \mathcal{M}(\kappa_q^T, y_{t-1}) \vee \big(\mathrm{argmax}(A_{t-1}) = q\big) \tag{12}$$

where $[\vee]$ is logical operator OR, and A_{t-1} denotes the attention score at $t-1$ time step. To preserve parallel training in transformer, we mask semantic kernel after its aligned target token (from external alignment tool) is generated at training.

3.4 Training Strategy

The overall loss function is divided into two parts: a translation loss $\mathcal{L}_D$ and an N-gram smoothing loss $\mathcal{L}_g$ for Projector. The overall loss function formulates:

$$\mathcal{L} = \mathcal{L}_D + \lambda \cdot \mathcal{L}_g \tag{13}$$

where $\lambda \in [0, 1]$ is a hyper-parameter to balance the impact between two losses. Details about N-gram smoothing loss can be found in Sec-3.2. After integrating semantic kernels, the translation loss is like:

$$\mathcal{L}_D = \sum_{\langle X,Y\rangle \in C} \sum_{y \in Y} \log P(y|y_{<t}, \mathcal{K}^T, X) \tag{14}$$

We set a norm threshold γ to control how strict we choose content words, explained in Sec-3.1. However, the norm calculation made at early stages is usually unreliable. We propose norm threshold annealing, which is computed as $e \cdot \gamma + (1 - e)$ where e is gradually annealed from 0 to 1 during the first 1/3 of training steps.

4 Experiments

We conduct experiments on the following benchmarks: NIST Chinese to English (Zh→En), WMT14 English to German (En→De), WMT14 English to French (En→Fr), IWSLT14 English to/from German (En↔De) translation tasks.

4.1 Datasets

For WMT 14 En→De, the training corpus is identical to previous work (Wang et al., 2019), which consists of about 4.5M sentence pairs. The validation set is newstest2013 and test set is newstest2014. For WMT 14 En→Fr, this dataset contains 36M sentences. The validation set is the concatenation of newstest2012 and newstest2013. Test results are reported on newstest2014 as (Wang et al., 2019). Following previous work (Yang et al., 2020), IWSLT 14 En→De dataset contains 160k sentence pairs for training and 7584 sentence pairs for validation. The concatenation of validation sets is used as the test set (dev2010, dev2012, tst2010, tst2011, tst2012). For NIST Zh→En, we use the LDC corpus with 1.25M sentence pairs with 27.9M Chinese words and 34.5M English words, respectively. We select the best model using the NIST 2002 as the validation set for model selection and hyperparameter tuning. The NIST 2004 (MT04), 2005 (MT05), 2006 (MT06) and 2008 (MT08) datasets are used as test sets.

We choose the Stanford segmenter (Tseng et al., 2005) for Chinese word segmentation and apply the script `tokenizer.pl` of Moses (Koehn et al., 2007) for English, French, and German tokenization. All data has been jointly byte pair encoded (BPE) (Sennrich et al., 2016). For WMT/IWSLT, we create a joint vocabulary with 32k and 10k merge operations respectively. For NIST Zh→En, BPEs are learnt separately with 60k operations.

	En→De	En→Fr	Zh→En	Params	Time Ratio ↓
Transformer (Vaswani et al., 2017)	28.40	41.80	-	213M	-
Transformer+Deli (Xia et al., 2017)	29.11^{+}	42.58^{+}	-	$372M^{+}$	1.79 ×
Soft-Prototype (Wang et al., 2019)	29.46	**42.99**	-	200.2M	1.35 ×
GNMT (Shah and Barber, 2018)	$28.81^{\dagger}$	$42.20^{\dagger}$	$46.69^{\ddagger}$	$289M^{*}$	2.08 ×
Mirror-GNMT (Zheng et al., 2020)	$29.22^{\dagger}$	-	46.98	$474M^{*}$	2.70 ×
SD-NMT (Ai and Fang, 2021)	29.49	**42.97**	-	-	2.44 ×
Transformer (our implementation)	28.55	41.84	45.88	214M	-
SKAM	**29.52**	**42.95**	**47.00**	252M	**1.20** ×

Table 1: Results on WMT14 En→De, WMT14 En→Fr and NIST Zh→En translation tasks. Results marked $^{+}$, ‡, † are from Wang et al. (2019); Zheng et al. (2020); Ai and Fang (2021), respectively. Numbers marked with * are from our implementation. "Params" denotes the number of model parameters for En→De. "Time Ratio" is calculated as the ratio of inference time between each model and transformer baseline.

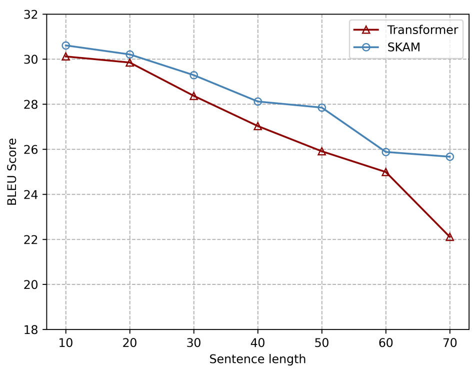

Figure 4: BLEU scores according to the sentence length. Results are on WMT14 En→De. Apparently, the longer the sentence, the better the performance that SKAM outperforms Transformer baseline.

We use GIZA++ (Och and Ney, 2003) as the external word alignment tool. As the whole model works on the sub-word level, following previous work (Chen et al., 2020b; Zenkel et al., 2020), we apply BPE units instead of words for alignment.

4.2 Model Configuration

Fundamental Transformer is implemented with `fairseq` (Ott et al., 2019). We follow the most common model configuration for each dataset. For IWSLT/NIST/WMT, we use the small/base/big transformer model. In detail, the encoder and decoder include 6 layers. All layers have an embedding size of 512/512/1024, a feed-forward size of 1024/2048/4096 and 4/8/16 attention heads, respectively. In order to prevent overfitting, we use a dropout rate of 0.3 (except for WMT 14 En→Fr, which is 0.1), and label smoothing of 0.1. For IWSLT and NIST, we train the model on a single P100 GPU, with each batch containing 4096 tokens. For WMT, we train the model on 6 P100 GPUs with update frequency set to 2, which results in $2500\times6\times2$ tokens per batch. We average the last 5/20 checkpoints for base/big model and use the checkpoint that has the best valid performance for small model. We use the case-sensitive tokenized BLEU `multi-bleu.perl` (Papineni et al., 2002) to evaluate WMT tasks and case-insensitive tokenized BLEU `mteval-v11b.pl` for NIST Zh→En. We report `sacrebleu` (Post, 2018) results for IWSLT. All experiments are run 4 times and report the average BLEU.

Projector is implemented as transformer encoder with 3 layers. The feed-forward size and attention heads are the same as fundamental transformer for each dataset. After adding projector, the training speed is on average about 80% of the vanilla transformer. For all benchmarks, we set $\lambda = 0.3$ heuristically. Norm threshold γ is set to 0.5 and $N = 3$ in our main experiment unless otherwise specified. We update adaptive mask with attention score from the top layer of decoder.

4.3 Baselines

For strictly consistent comparison, we involve the following strong baselines: **Transformer** (Vaswani et al., 2017) is a strong baseline which we build our model upon. **Deliberation Network** (Xia et al., 2017) and **SoftPrototype** (Wang et al., 2019) first generate the draft and polish the draft for the final translation. **GNMT** (Shah and Barber, 2018), **Mirror-GNMT** (Zheng et al., 2020) and **SD-NMT**

IWSLT14	En→De	De→En	Params	Avg.Δ
SKAM	**29.61**	**35.68**	43M	-
w/o $\mathcal{L}_g$	28.95	35.11	43M	-0.62
w/o AM	29.26	35.29	43M	-0.37
w/o $s(\cdot)$	29.02	35.21	43M	-0.53
Transformer	28.60	34.56	37M	-1.06

Table 2: Results on IWSLT14 En↔De translation tasks and Ablation Study. Avg.Δ means the gap between each model setting and SKAM. "w/o $s(\cdot)$" means the semantic kernels are selected randomly from source sentences.

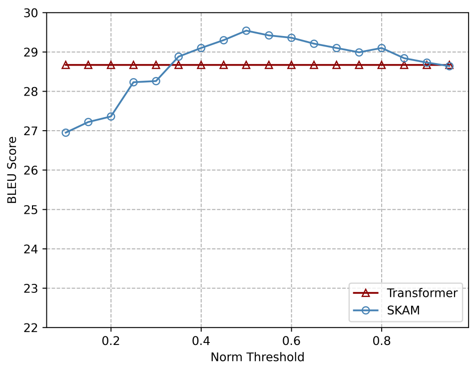

Figure 5: Test of different norm thresholds γ on IWSLT14 En→De. $\gamma = 0$ means that all source words are treated as semantic kernels, while $\gamma = 1$ indicates no semantic kernels are selected at all.

(Ai and Fang, 2021) sample a latent semantic embedding from semantic space and consider it as global information for decoding.

4.4 Results and Comparison

The results for WMT14 En →{De, Fr} and NIST Zh→En are presented in Table 1 and results on IWSLT14 En↔De are in Table 2. For convenience, we refer to our model as "SKAM" in these tables. We summarize the results as:

Semantic kernels improve model performance. Compared with transformer baseline, our approach on all four benchmarks brings substantial improvements, 1.07 BLEU points on average. Our model obtains competitive performance compared with previous methods on several benchmarks, and even surpasses all previous methods with a 29.52 BLEU score on WMT14 En-De benchmark. All results are statistically significant with $p < 0.01$ in paired bootstrap sampling (Koehn, 2004).

N-gram	0	1	3	5
SKAM	28.95	29.25	29.61	29.43

Table 3: Test of our N-gram smoothing supervision. The experiments are conducted on IWSLT14 En → De. $N = 0$ means no supervision is applied on Projector module.

Semantic kernels are time efficiency. As our semantic kernels are generated in a non-autoregressive way, our model only needs about 17% extra time to generate them. Compared with previous work, our model achieves about **1.7** times faster on average, even 2 times faster than some latent semantic-based methods.

4.5 Ablation Study

We perform an ablation study to show the effectiveness of each module on IWSLT14 En↔De benchmarks. The results are shown in Table 2. Specifically, "w/o $s(\cdot)$" compares our model with a baseline in which the decoder extends its K, V matrix with random parameters. Also, the results show that the improvements mainly come from our design, not an increase in parameters.

4.6 Parameter Analysis

Effect of Norm Threshold Norm threshold γ controls how strict we select semantic kernels. In general, the bigger γ is, the fewer words are selected as semantic kernels. To further examine the impact of norm threshold γ, we conduct experiments on IWSLT14 En→De benchmark. From the results, we find that when $\gamma < 0,5$, the performance increases, for we filter out more and more irrelevant words in expressing semantics. When $\gamma > 0.5$, performance gradually decreases and the model eventually deteriorates to transformer baseline.

Effect of N-gram We also test the impact of N-gram smoothing supervision on the Projector and depict the results in Table 3. Intuitively, the bigger N is, the better to disambiguate each word while the smaller N is, the better the discrepancy among each representation. From Table 3, we find that N-gram smoothing loss is critical to Projector and $N = 3$ is a balance point between the discrepancy and disambiguation.

Source	**So I want you** to **think** about a thought experiment .
Reference	**Daher möchte ich** , dass Sie über ein Gedankenexperiment **nachdenken** .
Transformer	**Denken** Sie also an ein Gedankenexperiment .
Keywords	**So**, **I**, **want**, **you**, **think**, thought, experiment
SKAM	**Ich möchte** , dass Sie über ein Gedankenexperiment **nachdenken** .
Source	" **Bottom line is that** with costs rising , people in the middle to **lower** end (of the income scale) will be looking to supplement their income wherever they can , " says Song Seng Wun , economist at CIMB , a Malaysian bank .
Reference	„ **Im Endeffekt bedeutet das** , dass angesichts steigender Kosten die Menschen im mittleren bis **unteren** Segment (der Einkommensskala) versuchen werden , ihr Einkommen zu ergänzen , wo immer das möglich ist “ , sagt Song Seng Wun , Ökonom bei CIMB , einer malaysischen Bank .
Transformer	„ Bei steigenden Kosten versuchen die Menschen in der Mitte bis **unten** (der Einkommensskala) , ihr Einkommen überall dort aufzubessern , wo sie können “ , sagt Song Seng Wun , Ökonom der malaysischen Bank CIMB .
Keywords	**Bottom**, **line**, costs, rising, people, middle, **lower**, end, income, scale, will, be, looking, supplement, their, income, wherever, they, can, says, Song, Seng, Wun, economist, at, CIMB, Malaysian, bank
SKAM	„ **Unterm Strich geht es darum** , dass die Menschen im mittleren bis **unteren** Bereich (der Einkommensskala) bei steigenden Kosten versuchen werden , ihr Einkommen zu erhöhen , wo immer sie können “ , sagt Song Seng Wun , Ökonom bei der CIMB , einer malaysischen Bank .

Table 4: Translation examples extracted from WMT 14 En→De task. “Keywords” denotes the words selected by our Norm-based Significance Score. The same color across different sentences refers to the same aligned sentence piece.

4.7 Performance w.r.t Sentence Length

Following previous work (Wang et al., 2019), we divide source sentences into different groups according to sentence length and compute the BLEU score separately for each group on WMT14 En→De task, as shown in Figure 4. Generally, the longer the source sentence is, the more influential semantic kernels are. This demonstrates that semantic kernels are especially helpful for the generation of longer sentences.

4.8 Case Study

We present examples from WMT 14 En→De task to illustrate the impact of semantic kernels, shown in Table 4, including source sentence, the gold target sentence (reference), translation generated by the vanilla Transformer model (Transformer) and translation given by ours (SKAM). From Table 4, we find that semantic kernels can help transformer baseline in two ways:

Select Words More Appropriately. In the first example, *nachdenken* is a more appropriate translation of *think* than *Denken* from Transformer. Similarly, in the second example, Transformer mistranslates *lower* into *unten* (bottom). We conjecture that the semantic kernels can help our model focus on meanings not word forms.

Capture Source Semantics More Comprehensively. In the first example, the sentence piece *So I want you* is missing by transformer, while SKAM successfully captures this meaning. This circumstance can also be found in the second example, where *Bottom line is that* is missing in transformer. This implies that SKAM is particularly helpful for the generation of longer and harder sentences. However, SKAM still shows some limitations. In the first example, the meaning *daher* (so) is missing in SKAM. More cases can be found in Appendix A.

5 Conclusion

Following Functional Equivalence Theory, we propose Semantic Kernels with Adaptive Decoding, which extracts several semantic kernels and projects them into target embedding space to guide translation. We propose adaptive mask mechanism to enable each decoding step to access target-side global information. Several empirical results reveal that our SKAM is both expressive in semantics and efficient in time.

Our way of representing kernel sentences in NMT is intuitive and simple. In future work, we would like to explore better methods to capture sentence semantics.

Limitations

As we tentatively give a successful implementation of leveraging Functional Equivalence Theory into Neural Machine Translation framework, such paradigm deserves a further and more detailed exploration. First, our representation of semantic kernels is quite intuitive and simple, how to align semantics between source and target languages is still challenging and thrilling, yet still in its fledgeless stage. Aside from it, while extensive experiments demonstrate that SKAM consistently improves translation quality, applying our approach on other language generation tasks will evaluate the effectiveness of our work in a more general way.

Acknowledgement

We thank all of the anonymous reviewers (during ARR Feb. and ARR Jun.) for the helpful comments. We also thank Xuchen Li and Shengchun Yang for their instructive suggestions and invaluable help. This work is supported by the National Natural Science Foundation of China (No.U21B2009).

References

Xi Ai and Bin Fang. 2021. Almost free semantic draft for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3931–3941, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1171–1179.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020a. Content word aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 358–364, Online. Association for Computational Linguistics.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020b. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Noam Chomsky. 2009. *Syntactic structures*. De Gruyter Mouton.

Bryan Eikema and Wilker Aziz. 2019. Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 124–141, Florence, Italy. Association for Computational Linguistics.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1380–1390, New Orleans, Louisiana. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018. Bag-of-words as target for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 332–338, Melbourne, Australia. Association for Computational Linguistics.

Eugene Albert Nida and Charles Russell Taber. 1982. *The theory and practice of translation*, volume 8. Brill Archive.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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, Brussels, Belgium. 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.

Harshil Shah and David Barber. 2018. Generative neural machine translation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1353–1362.

Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. 2018. Variational recurrent neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5488–5495. AAAI Press.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Copenhagen, Denmark. Association for Computational Linguistics.

Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Neural machine translation with soft prototype. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6313–6322.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1784–1794.

Jian Yang, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Ming Zhou. 2020. Improving neural machine translation with soft template prediction. In *Proceedings of the 58th Annual Meeting of the Association* *for Computational Linguistics*, pages 5979–5989, Online. Association for Computational Linguistics.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas. Association for Computational Linguistics.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5698–5705. AAAI Press.

Yang Zhao, Yining Wang, Jiajun Zhang, and Chengqing Zong. 2018. Phrase table as recommendation memory for neural machine translation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4609–4615. ijcai.org.

Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. 2019. Dynamic past and future 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 931–941, Hong Kong, China. Association for Computational Linguistics.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. 2020. Mirror-generative neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics*, 7:91–105.

A More Analysis on Norm-based Significance Score

To give a better view of what kinds of words are selected by Norm-based Siginificance Score and how these words affect translation progress, we sample more sentences from WMT14 En→De benchmark and present them in Table 6.

A.1 Words Selected by Norm-based Significance Score

In Table 6, we show the words selected by Norm-based Significance Score as "Keywords". As you can tell, our Norm-based Significance Score tends to select content words from source sentences. Though some prepositions and conjunctions are wrongly selected, most words selected by Norm-based Significance Score are content words.

A.2 Impact of Semantic Kernels

From Table 6, we can tell that before applying semantic kernels, some colored sentence pieces are not covered in translation results, while after applying semantic kernels, the translation results are more complete. Also, in the first two cases, applying semantic kernels further helps our model translate words more accurately. From the results, it is clear that semantic kernels help transformer model obtain a more comprehensive view of the source sentence.

En→De	WMT19	WMT20	WMT21
SKAM	**46.23**	**37.76**	**30.58**
Transformer	45.57	36.81	29.61

Table 5: Results on WMT19, WMT20, WMT21 En→De newstest benchmarks.

B More Results on WMT Benchmarks

We also report the results on WMT19, WMT20, WMT21 En→De newstest benchmarks. We build SKAM model upon Transformer Big baseline and the model is trained on 282M bilingual language pairs, which is the combination of all parallel data released by WMT21. All words are split into subword units with 40k merge operations. The model is trained on 8 V100 (16G) GPUs with a batch size of 48k tokens in total ($3000 \times 8 \times 2$). We trained 10 epochs and averaged the last 5 checkpoints. The results are reported in Table 5. SKAM outperforms transformer baseline with 0.86 BLEU score on average on these 3 benchmarks.

Source	The concept is not a **universal hit** .
Reference	Das Konzept ist kein **universeller Hit** .
Transformer	Das Konzept ist kein **Universalschlag** .
Keywords	concept, **universal**, **hit**
SKAM	Das Konzept ist kein **universeller Hit** .
Source	However , **speaking** the truth is not a crime .
Reference	Die Wahrheit zu **sagen** ist aber kein Verbrechen .
Transformer	Die Wahrheit ist jedoch kein Verbrechen .
Keywords	However, **speaking**, truth, crime
SKAM	Die Wahrheit zu **sagen** , ist jedoch kein Verbrechen .
Source	Whether **producing** soap , turning candles , felting or making silk , there is a suitable **activity** whatever your age .
Reference	Ob Seife **herstellen** , Kerzen drehen , filzen oder Seile fertigen , für jedes Alter ist das Passende dabei .
Transformer	Ob Seife , Kerzen drehen , Filzen oder Seidenherstellung – für jedes Alter ist etwas dabei .
Keywords	Whether, **producing**, soap, turning, candles, felting, or, making, milk, there, suitable, **activity**, whatever, age
SKAM	Ob Seife **herstellen** , Kerzen drehen , filzen oder Seide herstellen , in jedem Alter gibt es eine passende **Aktivität** .
Source	The backlog in the aerospace division was $ 32.9 billion as of September 30 , unchanged from **December 31** .
Reference	Der Auftragsbestand in der Luft- und Raumfahrtsparte betrug am 30. September 32,9 Milliarden Dollar und war damit gegenüber dem **31. Dezember** unverändert .
Transformer	Der Auftragsbestand des Geschäftsbereichs Luft- und Raumfahrt belief sich zum 30. September unverändert auf 32,9 Milliarden US-Dollar .
Keywords	backlog, aerospace, division, was, $, 32.9, billion, as, September, 30, unchanged, from, **December**, **31**
SKAM	Der Auftragsbestand im Geschäftsbereich Luft- und Raumfahrt belief sich zum 30. September auf 32,9 Milliarden US-Dollar , unverändert zum **31. Dezember** .
Source	**In addition** , visitors will have the **special opportunity** to get to know the open air museum on a carriage journey drawn by Black Forest Chestnut horses .
Reference	**Darüber hinaus** haben die Besucher die **besondere Gelegenheit** , das Freilichtmuseum während einer Kutschfahrt mit Schwarzwälder Füchsen kennenzulernen .
Transformer	Auf einer Kutschenfahrt mit Schwarzwaldkutschenpferden lernen die Besucher das Freilichtmuseum näher kennen .
Keywords	**addition**, visitors, will, **special**, **opportunity**, get, know, open, air, museum, on, carriage, journey, drawn, by, Black, Forest, Chestnut, horses
SKAM	**Darüber hinaus** haben die Besucher die **besondere Gelegenheit** , das Freilichtmuseum auf einer Kutschenfahrt von Schwarzwald-Kastanienpferden kennen zu lernen .
Source	Following the renovation , plastering and planting of trees in the old internal school yard , within the two wings of the 1912 school , **as a subsequent measure** the boundary wall , which is in need of refurbishment , must be renovated from the ground up within the foreseeable future .
Reference	Nach der Sanierung , Pflasterung und Baumbepflanzung des alten Schulinnenhofes innerhalb der beiden Seitenflügel der 1912 erbauten Schule muss in absehbarer Zeit als Folgemaßnahme , die sanierungsbedürftige Begrenzungsmauer von Grund auf saniert und auf neuen Unterbau gestellt werden .
Transformer	Nach der Renovierung , Verputzung und Bepflanzung des alten Schulhofs in den zwei Flügeln der Schule von 1912 muss die sanierungsbedürftige Grenzmauer in absehbarer Zeit von Grund auf erneuert werden .
Keywords	Following, renovation, plastering, planting, trees, old, internal, school, yard, within, two, wings, 1912, school, **as**, **subsequent**, **measure**, boundary, wall, which, need, refurbishment, must, be, renovated, from, ground, up, within, foreseeable, future
SKAM	Nach der Renovierung , Verputzung und Anpflanzung von Bäumen im alten Schulhof , innerhalb der beiden Flügel der Schule von 1912 , muss **als Folgemaßnahme** in absehbarer Zeit die renovierungsbedürftige Grenzmauer von Grund auf erneuert werden .

Table 6: Translation examples extracted from WMT 14 En→De task.“Keywords” denotes the words selected by our Norm-based Significance Score. Same color across different sentences refers to the same aligned sentence piece.