Guiding Neural Machine Translation with Semantic Kernels

Ping Guo\textsuperscript{1,2}, Yue Hu\textsuperscript{1,2,*}, Xiangpeng Wei\textsuperscript{3}, Yubing Ren\textsuperscript{1,2}, Yunpeng Li\textsuperscript{1,2}, Luxi Xing\textsuperscript{4}, Yuqiang Xie\textsuperscript{1,2}

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

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.

*Corresponding Author

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 “KS” and “KT” represent source and target semantic kernels, respectively. “Proj” is our projector.

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
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 sentence respectively, where $I$ and $J$ are the sentence lengths. Given a bilingual sentence pair $(X, Y)$, an NMT model learns a set of parameters $\Theta$ to maximize the posterior probability $P(Y|X; \Theta)$:

$$P(Y|X; \Theta) = \prod_{t=0}^{J} P(y_t|y_{<t}, X; \Theta)$$  \hspace{1cm} (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 $\text{ATT}(Q, K, V)$ is calculated as:

$$\text{ATT}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2)$$

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:

$$O^l_e = \text{ATT}(H^{l-1}_e, H^{l-1}_e, H^{l-1}_e) + H^{l-1}_e$$
$$H^l_e = \text{LN}(\text{FFN}(\text{LN}(O^l_e))) + \text{LN}(O^l_e) \quad (3)$$

where $\text{LN}()$ and $\text{FFN}()$ 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 $H^l_e$:

$$O^l_d = \text{ATT}(H^{l-1}_d, H^{l-1}_d, H^{l-1}_d) + H^{l-1}_d$$
$$S^l_d = \text{ATT}(\text{LN}(O^l_d), H^L_d, H^C_d) + \text{LN}(O^l_d)$$
$$H^l_d = \text{LN}(\text{FFN}(\text{LN}(S^l_d))) + \text{LN}(S^l_d) \quad (4)$$

where the top layer of the decoder $H^L_d$ 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:

$$\mathcal{K}^S(X) := \{\text{ENC}(x_i|X) \mid s(x_i) > 0, \ x_i \in X\} \quad (5)$$

where $\text{ENC}$ 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-
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 \( \max_{v \in \mathcal{V}_S} (||v||) \) in source embedding. Our significance score \( s(\cdot) \) is formulated as:

\[
s(x_i) = \frac{||x_i||}{\max_{v \in \mathcal{V}_S} (||v||)} - \gamma
\]

where \( \gamma \in [0, 1] \) is a norm threshold value. We only choose words whose score \( s(x_i) \) is larger than \( \gamma \) 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 \( \mathcal{K}^T \):

\[
\mathcal{K}^T = f_{S \rightarrow T}(\mathcal{K}^S)
\]

where \( f_{S \rightarrow T} \) 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 \( \text{ENC}(x_i | X) \), 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,

\[
\text{Span}(y_i) = \{ \tilde{y}_{i-k}, \tilde{y}_{i-k+1}, \ldots, \tilde{y}_i, \ldots, \tilde{y}_{i+k-1}, \tilde{y}_{i+k} \}
\]

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 \( \text{ENC}(x_i | X) \) as input. The \( N \)-gram smoothing loss \( L_g \) for one sample \( X \) formulates:

\[
L_g = \sum_{x_i \in X} \frac{1}{N} \sum_{m=0}^{2k} \log P(\tilde{y}_{i+m} | \text{ENC}(x_i | X))
\]

(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.

\[
\text{ATT}(H^{l-1}_d, [\mathcal{K}^T; H^{l-1}_d], [\mathcal{K}^T; H^{l-1}_d])
\]

(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.
Adaptive Mask

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

$$
M(k^T_q, y_t) = \begin{cases} 
0, & k^T_q \text{ is not contained in } y_{<t} \\
1, & k^T_q \text{ is contained in } y_{<t} 
\end{cases}
$$

(11)

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

$$
M(k^T_q, y_t) = M(k^T_q, y_{t-1}) \lor (\text{argmax}(A_{t-1}) = q)
$$

(12)

where $[\lor] \text{ is logical operator OR, and } A_{t-1} \text{ denotes the attention score at } t-1 \text{ 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 $L_D$ and an $N$-gram smoothing loss $L_g$ for Projector. The overall loss function formulates:

$$
L = L_D + \lambda \cdot L_g
$$

(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:

$$
L_D = \sum_{(X,Y) \in C} \sum_{y \in Y} \log P(y | y_{<t}, K^T, X)
$$

(14)

We set a norm threshold $\gamma$ 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.
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.

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

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 2500×6×2 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 $\gamma$ 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...
Table 2: Results on IWSLT14 En→De translation tasks and Ablation Study. Avg.∆ means the gap between each model setting and SKAM. “w/o s(·)” means the semantic kernels are selected randomly from source sentences.

Table 3: Test of our N-gram smoothing supervision. The experiments are conducted on IWSLT14 En→De benchmark. 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(·)” 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 γ < 0.5, the performance increases, for we filter out more and more irrelevant words in expressing semantics. When γ > 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.
So I want you to think about a thought experiment.

Daher möchte ich, dass Sie über ein Gedankenexperiment nachdenken.

Denken Sie auch an ein Gedankenexperiment.

So, I, want, you, think, thought, experiment

Ich möchte, dass Sie über ein Gedankenexperiment nachdenken.

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.

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.

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.

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

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 malaysischen Bank CIMB.

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


A More Analysis on Norm-based Significance Score

To give a better view of what kinds of words are selected by Norm-based Significance 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.

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 sub-word units with 40k merge operations. The model is trained on 8 V100 (16G) GPUs with a batch size of 48k tokens in total (3000 × 8 × 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>WMT19</th>
<th>WMT20</th>
<th>WMT21</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKAM</td>
<td>46.23</td>
<td>37.76</td>
<td>30.58</td>
</tr>
<tr>
<td>Transformer</td>
<td>45.57</td>
<td>36.81</td>
<td>29.61</td>
</tr>
</tbody>
</table>

Table 5: Results on WMT19, WMT20, WMT21 En→De newstest benchmarks.
<table>
<thead>
<tr>
<th>Source</th>
<th>The concept is not a universal hit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Das Konzept ist kein universeller Hit.</td>
</tr>
<tr>
<td>Transformer</td>
<td>Das Konzept ist kein Universalschlag.</td>
</tr>
<tr>
<td>Keywords</td>
<td>concept, universal, hit</td>
</tr>
<tr>
<td>SKAM</td>
<td>Das Konzept ist kein universeller Hit.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>However, speaking the truth is not a crime.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Die Wahrheit zu sagen ist aber kein Verbrechen.</td>
</tr>
<tr>
<td>Transformer</td>
<td>Die Wahrheit ist jedoch kein Verbrechen.</td>
</tr>
<tr>
<td>Keywords</td>
<td>However, speaking, truth, crime</td>
</tr>
<tr>
<td>SKAM</td>
<td>Die Wahrheit zu sagen, ist jedoch kein Verbrechen.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Whether producing soap, turning candles, felting or making silk, there is a suitable activity whatever your age.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Ob Seife herstellen, Kerzen drehen, filzen oder Seile fertigen, für jedes Alter ist das Passende dabei.</td>
</tr>
<tr>
<td>Transformer</td>
<td>Ob Seife , Kerzen drehen, Filzen oder Seidenherstellung – für jedes Alter ist etwas dabei.</td>
</tr>
<tr>
<td>Keywords</td>
<td>Whether, producing, soap, turning, candles, felting, or, making, milk, there, suitable, activity, whatever, age</td>
</tr>
<tr>
<td>SKAM</td>
<td>Ob Seife herstellen, Kerzen drehen, filzen oder Seide herstellen, in jedem Alter gibt es eine passende Aktivität.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>The backlog in the aerospace division was $32.9 billion as of September 30, unchanged from December 31.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>backlog, aerospace, division, was, $, 32.9, billion, as, September, 30, unchanged, from, December, 31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Darüber hinaus haben die Besucher die besondere Gelegenheit, das Freilichtmuseum während einer Kutschfahrt mit Schwarzwälder Füchsen kennenzulernen.</td>
</tr>
<tr>
<td>Transformer</td>
<td>Auf einer Kutschenfahrt mit Schwarzwaldkutschenpferden lernen die Besucher das Freilichtmuseum näher kennen.</td>
</tr>
<tr>
<td>Keywords</td>
<td>addition, visitors, will, special, opportunity, get, know, open, air, museum, on, carriage, journey, drawn, by, Black, Forest, Chestnut, horses</td>
</tr>
<tr>
<td>SKAM</td>
<td>Darüber hinaus haben die Besucher die besondere Gelegenheit, das Freilichtmuseum auf einer Kutschenfahrt von Schwarzwald-Kastanienpferden kennen zu lernen.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Nach der Sanierung, Pflasterung und Baumbeplanzung 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.</td>
</tr>
<tr>
<td>Keywords</td>
<td>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</td>
</tr>
</tbody>
</table>

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.