Abstract

The integration of syntactic structures into Transformer machine translation has shown positive results, but to our knowledge, no work has attempted to do so with semantic structures. In this work we propose two novel parameter-free methods for injecting semantic information into Transformers, both rely on semantics-aware masking of (some of) the attention heads. One such method operates on the encoder, through a Scene-Aware Self-Attention (SASA) head. Another on the decoder, through a Scene-Aware Cross-Attention (SACrA) head. We show a consistent improvement over the vanilla Transformer and syntax-aware models for four language pairs. We further show an additional gain when using both semantic and syntactic structures in some language pairs.

1 Introduction

It has long been argued that semantic representation can benefit machine translation (Weaver, 1955; Bar-Hillel, 1960). Moreover, RNN-based neural machine translation (NMT) has been shown to benefit from the injection of semantic structure (Song et al., 2019; Marcheggiani et al., 2018). Despite these gains, to our knowledge, there have been no attempts to incorporate semantic structure into NMT Transformers (Vaswani et al., 2017). We address this gap, focusing on the main events in the text, as represented by UCCA (Universal Cognitive Conceptual Annotation; Abend and Rappoport, 2013), namely scenes.

UCCA is a semantic framework originating from typological and cognitive-linguistic theories (Dixon, 2009, 2010, 2012). Its principal goal is to represent some of the main elements of the semantic structure of the sentence while disregarding its syntax. Formally, a UCCA representation of a text is a directed acyclic graph where leaves correspond to the words of the sentence and nodes correspond to semantic units. The edges are labeled by the role of their endpoint in the relation corresponding to their starting point (see Fig. 1). One of the motivations for using UCCA is its capability to separate the sentence into “Scenes”, which are analogous to events (see Fig. 1). Every such Scene consists of one main relation, which can be either a Process (i.e., an action), denoted by P, or a State (i.e., continuous state), denoted by S. Scenes also contain at least one Participant (i.e., entity), denoted by A. For example, the sentence in Fig. 1 comprises two scenes: the first one has the Process ”saw” and two Participants – ”I” and ”the dog”; the second one has the Process ”barked” and a single Participant – ”dog”.

So far, to the best of our knowledge, the only structure-aware work that integrated linguistic knowledge and graph structures into Transformers used syntactic structures (Strubell et al., 2018; Bugliarello and Okazaki, 2020; Akoury et al., 2019; Sundararaman et al., 2019; Choshen and Abend, 2021). The presented method builds on the method proposed by Bugliarello and Okazaki (2020), which utilized a Universal Dependencies graph (UD; Nivre et al., 2016) of the source sentence to focus the encoder’s attention on each token’s parent, namely the token’s immediate ancestor in the UD graph. Similarly, we use the UCCA graph of the source sentence to generate a scene-aware mask for the self-attention heads of the encoder. We call this method SASA (see §2.1).

We test our model (§2) on translating English into four languages. Two that are more syntactically similar to English –(Nikolaev et al., 2020; Dryer and Haspelmath, 2013) German (En-De), Russian (En-Ru), and two that are much less so – Turkish (En-Tr) and Finnish (En-Fi). We find consistent improvements across multiple test sets for all four cases. In addition, we create a syntactic variant of our semantic model for better comparability. We observe that on average, our semanti-


Figure 1: Example of UCCA-parsing of the sentence "I saw the dog that barked" and its separation into two scenes.

Figur

cally aware model outperforms the syntactic models. Moreover, for the two languages less similar to English (En-Tr and En-Fi), combining both the semantic and the syntactic data results in a further gain. While improvements are often small, at times the combined version outperforms SASA and UD-ISCAL (our syntactic variant, see §3) by 0.52 and 0.69 BLEU points (or 0.46 and 0.43 chrF), respectively.

We also propose a novel method for introducing the source graph information during the decoding phase, namely through the cross-attention layer in the decoder (see §2.2). We find that it improves over the baseline and syntactic models, although SASA is generally better. Interestingly, for En-Fi, this model also outperforms SASA, suggesting that some language pairs may benefit more from semantic injection into the decoder.

Overall, through a series of experiments (see §4), we show the potential of semantics as an aid for NMT. We experiment with a large set of variants of our method, to see where and in what incorporation method they best help. Finally, we show that semantic models outperform UD baselines and can be complementary to them in distant languages, showing improvement when combined.

2 Models

Transformers have been shown to struggle when translating some types of long-distance dependencies (Choshen and Abend, 2019; Bisazza et al., 2021a) and when facing atypical word order (Bisazza et al., 2021b). Sulem et al. (2018a) proposed UCCA based preprocessing at inference time, splitting sentences into different scenes. They hypothesized that models need to decompose the input into scenes implicitly, and provide them with such a decomposition, as well as with the original sentence. They show that this may facilitate machine translation (Sulem et al., 2020) and sentence simplification (Sulem et al., 2018b) in some cases.

Motivated by these advances, we integrate UCCA to split the source into scenes. However, unlike Sulem et al., we do not alter the sentence length in pre-processing, as this method allows less flexibility in the way information is passed, and also re-implementing it yielded inferior results (see §A.5). Instead, we investigate ways to integrate the split into the attention architecture.

We follow previous work (Bugliarello and Okazaki, 2020) in the way we incorporate our semantic information. In their paper, Bugliarello and Okazaki (2020) introduced syntax in the form of a parent-aware mask, which was applied before the softmax layer in the encoder’s self-attention. We mask in a similar method to introduce semantics. However, parent in the UCCA framework is an elusive concept. Hence, we use a different way to express the semantic information in our mask, i.e., we make it scene-aware, rather than parent-aware.

Following Sulem et al. (2018b), we divide the source sentence into scenes, using the sentence’s UCCA parse. We then define our Scene-Aware mask

\[ M[i,j] = \begin{cases} 1, & \text{if } i,j \text{ in the same scene} \\ 0, & \text{otherwise} \end{cases} \]

Intuitively, an attention head masked this way is allowed to attend to other tokens, as long as they share a scene with the current one.\(^1\)

Our base model is the Transformer (Vaswani et al., 2017), which we enhance by making the attention layers more scene-aware. We force one\(^2\) of the heads to attend to words in the same scene which we assume are more likely to be related than words from different scenes. As we replace regular self-attention heads with our scene-aware ones, we maintain the same number of heads and layers as in the baseline.

2.1 Scene-Aware Self-Attention (SASA)

Figure 2 presents the model’s architecture. For a source sentence of length \( L \), we obtain the

\(^1\)In case a token belongs to more than one scene, as is the case with the word "dog" in Fig. 1, we allow it to attend to tokens of all the scenes it belongs to.

\(^2\)Initial trials with more than one head did not show further benefit for UCCA based models.
Figure 2: Scene-aware self-attention head for the input sentence "I saw the dog that barked", consisting of two scenes: "I saw the dog" and "dog barked".

keys, queries, and values matrices denoted by \( K^i, Q^i, V^i \in \mathbb{R}^{L \times d} \), respectively. We then multiply \( K^i \) with \( Q^i \), scale the result and pass it through a softmax layer. The difference between our method and a vanilla Transformer (Vaswani et al., 2017) is that the vanilla model multiplies the output of the softmax by \( V^i \). We, however, first mask it with our pre-generated scene-aware mask \( M^i_S \in \{0, 1\}^{L \times L} \), using a pointwise multiplication, and only then do we multiply the result with \( V^i \).

2.2 Scene-Aware Cross-Attention (SACrA)

Next, we design a model in which we integrate information about the scene structure through the cross-attention layer in the decoder (see Fig. 3). Thus, instead of affecting the overall encoding of the source, we bring forward the splits to aid in selecting the next token.

Formally, for a source sentence of length \( L_{\text{src}} \) and target sentence of length \( L_{\text{trg}} \), we compute for each head the queries and values matrices, denoted by \( Q^i \in \mathbb{R}^{L_{\text{trg}} \times d_{\text{model}}} \) and \( V^i \in \mathbb{R}^{L_{\text{src}} \times d} \), accordingly. Regarding key values, denoted by \( \tilde{K}^i \in \mathbb{R}^{L_{\text{src}} \times L_{\text{trg}}} \), we multiply the encoder’s output \( X_{\text{enc}} \in \mathbb{R}^{L_{\text{src}} \times d_{\text{model}}} \) with our pre-generated mask \( M^i_S \in \{0, 1\}^{L_{\text{src}} \times L_{\text{src}}} \), and then scale it by multiplying the result with \( \frac{1}{L_{\text{src}}} \). Finally, we pass \( V^i, Q^i \) and \( \tilde{K}^i \) through a regular attention layer, as with the standard Transformer architecture.

Scene-Aware Key Matrix. The rationale behind the way we compute our scene-aware keys matrix lies in the role of the keys matrix in an attention layer. In the cross-attention layer, the queries come from the decoder. Source-side contextual information is encoded in the keys, which come from the encoder. Therefore, when we assign the same scene masks to all the words that are included in the same set of scenes, the key values for these words will be the same, and they will thus be treated similarly by the query. As a result, the query will give the same weight to source tokens that share the same set of scenes. Therefore, a complete scene (or a few scenes), rather than specific tokens (as with the vanilla Transformer), will influence what the next generated token will be, which will in turn yield a more scene-aware decoding process.

3 Experimental Setting

Data Preparation. First, we unescaped HTML characters and tokenized all our parallel corpora (Koehn et al., 2007). Next, we removed empty sentences, sentences longer than 100 tokens (either on the source or the target side), sentences with a source-target ratio larger than 1.5, sentences that do not match the corpus’s language as determined by langid Lui and Baldwin, 2012, and sentences that fast align (Dyer et al., 2013) considers unlikely to align (minimum alignment score of -180). Then, for languages with capitalization, we trained true-casing models on the train set (Koehn et al., 2007) and applied them to all inputs to the network. Finally, we trained a BPE model (Sennrich et al., 2016), jointly for language pairs with a similar writing system (e.g., Latin, Cyrillic, etc.)
and separately otherwise, and then applied them accordingly.

We trained our model on the full WMT16 dataset for the English–German (En-De) task, using the WMT newstest2013 as development set. We also trained our models on a train set consisting of Yandex Corpus, News Commentary v15, and Wikititles v2 for the English–Russian (En-Ru) task. In addition, we trained our models on the full WMT19 dataset (excluding Paracrawl, in order to avoid noisiness in the data) for the English–Finnish (En-Fi). Finally, we trained on the full WMT18 dataset for the English–Turkish (En-Tr) task. For the test sets, we used all the newstests available for every language pair since 2012, excluding the one designated for development.

Models. Hyperparameters shared by all models are described in §3. We tune the number of heads that we apply the mask to (\#heads) and the layers of the encoder we apply SASA to (layer), using the En-De development set. We start with tuning the layers for SASA, which we find is layer = 4, and then we tune the \#heads (while fixing layer = 4), and get \#head = 1. We also use the En-De development set to tune the \#heads and the layers of the SACrA model in a similar fashion, namely first the layers and then the \#heads (with the tuned layers fixed). We find the best hyperparameters are \#heads = 1 and layers = 2&3. For both models, we apply the tuned hyperparameters to all other language pairs. Interestingly, while it is common practice to change all the layers of the model, we find it suboptimal. Moreover, the fact that semantic information is more beneficial in higher layers, in contrast to the syntactic information that is most helpful when introduced in lower layers (see §3) may suggest that semantics is relevant for more complex generalization, which is reminiscent of findings by previous work (Tenney et al., 2019a; Belinkov, 2018; Tenney et al., 2019b; Peters et al., 2018; Blevins et al., 2018; Slobodkin et al., 2021).

UCCA parses are extracted using a pretrained BERT-based TUPA model, that was trained on sentences in English, German and French (Hershcovich et al., 2017).

Binary Mask. For the SASA model, we experiment with two types of masks: a binary mask, as described in §2, and scaled masks, i.e.,

\[ M_{C[i,j]} = \begin{cases} 1, & \text{if } i,j \text{ in the same scene} \\ C, & \text{otherwise} \end{cases} \tag{1} \]

where \( C \in (0, 1) \). By doing so, we allow some out-of-scene information to pass through, while still emphasizing the in-scene information (by keeping the value of \( M \) for same-scene tokens at 1). In order to tune \( C \), we performed a small grid search over \( C \in \{0.05, 0.1, 0.15, 0.2, 0.3, 0.5\} \).

Additionally, similarly to Bugliarello and Okazaki (2020), we test a normally-distributed mask, according to the following equation:

\[ M_{i,j} = f_{\text{norm}}(x = C \cdot \text{dist}(i,j)) \tag{2} \]

where \( f_{\text{norm}} \) is the density function of the normal distribution:

\[ f_{\text{norm}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{3} \]

We define a scene-graph where nodes are scenes and edges are drawn between scenes with overlapping words. dist(i,j) is the shortest distance between tokens i and j. \( \sigma = \frac{1}{\sqrt{2\pi}} \), to ensure the value of \( M \) is 1 for words that share a scene (dist(i,j)=0), and \( C \) is a hyperparameter, which is determined through a grid search over \( C \in \{0.1, 0.2, 0.5, \sqrt{0.5}\} \). For each of those two scaled versions of the mask, we choose the mask which has the best performance and compare it to the binary mask (see 1). We find that neither outperforms the binary mask. Therefore, we report the rest of our experiments with the binary mask.

Baselines. We compared our model to a few other models:

- **Transformer.** Standard Transformer-based NMT model, using the standard hyperparameters, as described in §3.

- **PASCAL.** Following Bugliarello and Okazaki (2020), we generate a syntactic mask for the self-attention layer in the encoder. We extract a UD-graph (Nivre et al., 2016) with udpipe (Straka and Straková, 2017). The value of the entries of the masks equal (see equation 3):

\[ M_{p_{i,j}} = f_{\text{norm}}(x = (j - p_{i})) \tag{4} \]
Table 1: BLEU scores for the top versions of our binary mask, scaled mask, and normally-distributed mask methods across all the WMT En-De newstests. Each column contains the BLEU scores over the WMT newstest corresponding to the year the column is labeled with (e.g., the scores under column 2015 are for En-De newstest2015). For newstest2020, there was more than one version on WMT, each translated by a different person. Both versions were included, with the second version denoted with a “B”. The best score for each test set is boldfaced, unless none is better than the baseline Transformer.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>17.60</td>
<td>20.49</td>
<td>20.55</td>
<td>22.17</td>
<td>25.46</td>
<td>19.70</td>
<td>28.01</td>
<td>26.84</td>
<td>17.71</td>
<td>16.94</td>
</tr>
<tr>
<td>+ binary mask</td>
<td>17.64</td>
<td>20.37</td>
<td>20.84</td>
<td>22.48</td>
<td>25.32</td>
<td>19.76</td>
<td>28.36</td>
<td>26.80</td>
<td>17.74</td>
<td>16.98</td>
</tr>
<tr>
<td>(n=1, l=4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ scaled mask</td>
<td>17.41</td>
<td>20.21</td>
<td>20.53</td>
<td>22.43</td>
<td>24.95</td>
<td>19.81</td>
<td>28.25</td>
<td>27.21</td>
<td>18.03</td>
<td>17.01</td>
</tr>
<tr>
<td>(n=2, l=4, C=0.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ normally distributed mask</td>
<td>17.39</td>
<td>20.52</td>
<td>20.57</td>
<td>22.24</td>
<td>25.44</td>
<td>19.63</td>
<td>28.35</td>
<td>26.60</td>
<td>17.14</td>
<td>16.77</td>
</tr>
<tr>
<td>(n=2, l=4, C=√(0.5))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with \( \sigma = 1 \) and \( p_t \) being the middle position of the \( t \)-th token’s parent in the UD graph of the sentence.

We use the same general hyperparameters as in the Transformer baseline. In addition, following the tuning of Bugliarello and Okazaki (2020), we apply the PASCAL mask to five heads of the first attention layer of the encoder, but unlike the original paper, we apply it after the layer’s softmax, as it yields better results and also resembles our model’s course of action.

- **UDISCAL.** In an attempt to improve the PASCAL model, we generate a mask that instead of only being sensitive to the dependency parent, is sensitive to all the UD relations in the sentences. We denote it UD-Distance-Scaled mask (UDISCAL). Namely, in order to compute the mask, we use a similar equation to that of PASCAL, with a minor alteration:

\[
M_{i,j} = f_{\text{norm}}(x = \text{dist}(i, j))
\]  

(5)

Where \( \sigma = 1 \), and \( \text{dist}(i, j) \) is defined to be the distance between the token \( i \) and the token \( j \) in the UD graph of the sentence while treating the graph as undirectional. As with the PASCAL layer, we apply the UD-scaled mask after the softmax layer. But, unlike the PASCAL head, we tuned the architecture’s hyperparameters to be just one head of the first layer, after performing a small grid search, namely testing with all layers \( l \in [1, 4] \), and then with \( \# \text{head} \in [1, 5] \).

**Training Details.** All our models are based on the standard Transformer-based NMT model (Vaswani et al., 2017), with 4000 warmup steps. In addition, we use an internal token representation of size 256, per-token cross-entropy loss function, label smoothing with \( \epsilon_l = 0.1 \) (Szegedy et al., 2016), Adam optimizer, Adam coefficients \( \beta_1 = 0.9 \) and \( \beta_2 = 0.98 \), and Adam \( \epsilon = e^{-1} \). Furthermore, we incorporate 4 layers in the encoder and 4 in the decoder, and we employ a beam-search during inference, with beam size 4 and normalization coefficient \( \alpha = 0.6 \). In addition, we use a batch size of 128 sentences for the training. We use \( \text{chrF}++ \) \( \text{py} \) with 1 word and beta of 3 to obtain \( \text{chrF}++ \) (Popovic, 2017) score as in WMT19 (Ma et al., 2019) and detokenized BLEU (Papineni et al., 2002) as implemented in Moses. We use the Nematus toolkit (Sennrich et al., 2017), and we train all our models on 4 NVIDIA GPUs for 150K steps. The average training time for the vanilla Transformer is 21.8 hours, and the average training time for the SASA model is 26.5 hours.

4 Experiments

We hypothesize that NMT models may benefit from the introduction of semantic structure, and present a set of experiments that support this hypothesis using the above-presented methods.

4.1 Scene-Aware Self-Attention

We find that on average, SASA outperforms the Transformer for all four language pairs (see 3), at times having gains larger than 1 BLEU point. Moreover, we assess the consistency of SASA’s gains, using the sign-test, and get a p-value smaller than
<table>
<thead>
<tr>
<th>Source sentences and Translations</th>
<th>Literal Translations into English</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC I promised a show?</td>
<td>BASE Я обещал показать?</td>
</tr>
<tr>
<td>SASA Я обещал шоу?</td>
<td>I promised a show?</td>
</tr>
<tr>
<td>SRC students said they looked forward to his class.</td>
<td>BASE Студенты сказали, что они с нетерпением ждали своего класса.</td>
</tr>
<tr>
<td>SASA Студенты сказали, что они с нетерпением ждали своего класса.</td>
<td>I remember those kids, with which I played in yard, that never got out (“that” and &quot;got out&quot; refer to kids).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Examples of correct translations generated by SASA, compared to the baseline Transformer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01, thus exhibiting a statistically significant improvement (see §A.4). We see a similar trend when evaluating the performance using the chrF metric (see §A.2), which further highlights our model’s consistent gains.</td>
</tr>
<tr>
<td>We also evaluate our model’s performance on sentences with long dependencies (see A.3), which were found to pose a challenge for Transformers (Choshen and Abend, 2019). We assume that such cases could benefit greatly from the semantic introduction. In contrast to our hypothesis, we find the gain to be only slightly larger than in the general case, which leads us to conclude the improvements we see do not specifically originate from the syntactic challenge. Nevertheless, we still observe a consistent improvement, with gains of up to 1.41 BLEU points, which further underscores our model’s superiority over the baseline model.</td>
</tr>
</tbody>
</table>

**Qualitative Analysis.** Table 2 presents a few examples in which the baseline Transformer errs, whereas our model translates correctly. In the first example, the Transformer translates the word “show” as a verb, i.e. *to show*, rather than as a noun. In the second example, the baseline model makes two errors: it misinterprets the word "look forward to" as "look at", and it also translates it as a present-tense verb rather than past-tense. The third example is particularly interesting, as it highlights our model’s strength. In this example, the Transformer makes two mistakes: first, it translates the part "play with (someone) in the yard" as "play with the yard". Next, it attributes the descriptive clause "which never got out" to the yard, rather than to the children. It seems then that introducing information about the scene structure into the model facilitates the translation, since it both groups the word "kids" with the phrase "I used to play with in the yard", and it also separates "never got out" from the word "yard". Instead, it clusters the latter with "kids", thus highlighting the relations between words in the sentence. In general, all these examples are cases where the network succeeds in disambiguating a word in its context. |

**4.2 Comparison to Syntactic Masks**

Next, we wish to compare our model to other baselines. Given that this is the first work to incorporate semantic information into the Transformer-based NMT model, we compare our work to syntactically-infused models (as described in §3): one is the PASCAL model (Bugliarello and Okazaki, 2020), and the other is our adaptation of PASCAL, the UD-Distance-Scaled (UDISCAL) model, which resembles better our SASA mask. We find (Table 3) that on average, SASA outperforms both PASCAL and UDISCAL. We also compare SASA with each of the syntactic models, finding that it is significantly (sign-test $p < 0.01$; see §A.4) better. This suggests that semantics might be more beneficial for Transformers than syntax.
Table 3: BLEU scores for the baseline Transformer model, previous work that used syntactically infused models – PASCAL and UDICAL, our SASA and SACrA models, and models incorporating UDICAL with SASA or SACrA, across all WMT’s newstests. For every language pair, each column contains the BLEU scores over the WMT newestst corresponding to the year the column is labeled with (e.g., for En-Ru, the scores under column 2015 are for En-Ru newstest2015). For some newstests, there was more than one version on WMT, each translated by a different person. For those test sets, we included both versions, denoting the second one with a “B”. In addition, for every language pair, the right-most column represents the average BLEU scores over all the pair’s reported neweststs. For every test set (and for the average score), the best score is boldfaced. For each of the semantic models (i.e., SASA and SACrA), improvements over all the baselines (syntactic and Transformer) are reported newstests. For every test set, each mask individually are marked with an asterisk. For models with both syntactic and semantic masks, improvements over each mask individually are marked with an asterisk. For those test sets, we included both versions, denoting the second one with a “B”.
4.3 Combining Syntax and Semantics

Naturally, our next question is whether combining both semantic and syntactic heads will further improve the model’s performance. Therefore, we test the combination of SASA with either PASCAL or UDISCAL, retaining the hyperparameters used for the separate models. We find that combining with UDISCAL outperforms the former, and so we continue with it. Interestingly, En-De and En-Ru hardly benefit from the combination compared just to the SASA model. We hypothesize that this might be due to the fact that the syntax of each language pair is already quite similar, and therefore the model mainly relies on it to separate the sentence that UCCA gives it as well. On the other hand, En-Fi and En-Tr do benefit from the combination, both on average and in most of the test sets.

Evaluating the performance using the chrF metric (see A.2) yields a similar behavior, which further confirms its validity. It leads us to hypothesize that language pairs that are more typologically distant from one another can benefit more from both semantics and syntax; we defer a more complete discussion of this point to future work. In order to confirm that the combined version persistently outperforms each of the separate versions for typologically distant languages, we compare each of the pairs using the sign-test (only on the test sets of En-Fi and En-Tr). We get a p-value of 0.02 for the comparison with SASA and 0.0008 for the comparison with UDISCAL. This suggests that for these language pairs, there is indeed a significant benefit, albeit small, from the infusion of both semantics and syntax.

4.4 Scene-Aware Cross-Attention

Following the analysis on the scene-aware self-attention, we wish to examine whether Transformers could also benefit from injecting source-side semantics into the decoder. For that, we develop the Scene-Aware Cross-Attention (SACrA) model, as described in §2.2. Table 3 presents the results of SACrA, compared to the Transformer baseline and SASA. We find that in general SASA outperforms SACrA, suggesting that semantics is more beneficial during encoding. With that said, for three out of the four language pairs, SACrA does yield gains over the Transformer, albeit small, and for one language pair (En-Fi) it even outperforms SASA on average. Moreover, comparing SACrA to the Transformer using the sign-test (see A.4) shows significant improvement ($p = 0.047$).

Surprisingly, unlike its self-attention counterpart, combining the SACrA model with UDISCAL does not seem to be beneficial at all, and in most cases is even outperformed by the baseline Transformer. We hypothesize that this occurs because appointing too many heads for our linguistic injection is inefficient when those heads cannot interact with each other directly, as the information from the UDISCAL head reaches the SACrA head only after the encoding is done. One possible direction for future work would be to find ways to syntactically enrich the decoder, and then to combine it with our SACrA model.

5 Conclusion

In this work, we suggest two novel methods for injecting semantic information into an NMT Transformer model – one through the encoder (i.e. SASA) and one through the decoder (i.e. SACrA). The strength of both methods is that they both do not introduce more parameters to the model, and only rely on UCCA-parses of the source sentences, which are generated in advance using an off-the-shelf parser, and thus do not increase the complexity of the model. We compare our methods to previously developed methods of syntax injection, and to our adaptation to these methods, and find that semantic information tends to be significantly more beneficial than syntactic information, mostly when injected into the encoder (SASA), but at times also during decoding (SACrA). Moreover, we find that for distinct languages, adding both syntax and semantics further improves the performance of the translation models. Future work will further investigate the benefits of semantic structure in Transformers, alone and in unison with syntactic structure.

References


Maja Popovic. 2017. chrf++: words helping character n-grams. In WMT.


A Appendix

A.1 Layer Hyperparameter-tuning for SASA

In order to optimize the contribution of the SASA model, we tuned the hyperparameter of the best layers in the encoder to incorporate our model, using the En-De newstest2013 as our development set. Table 4 presents the results.

A.2 ChrF Results

In order to reaffirm our results, we also evaluate the performance of all the models using the chrF metric (see §7). Indeed, all the different behaviors and trends we observed when evaluating using the Bleu metric (see §4) seem to be preserved when under the chrF metric. This further validates our results.

A.3 Challenge Sets

In addition to testing on the full newstests sets, we also experiment with sentences characterized by long dependencies, which were shown to present a challenge for Transformers (Choshen and Abend, 2019). In order to acquire those challenge sets, we use the methodology described by Choshen and Abend (2019), which we apply on each of the newstest sets. In addition, for the En-Tr task, which has a limited number of newstests, we generate additional challenge sets, extracted from corpora downloaded from the Opus Corpus engine (Tiedeman, 2012): the Wikipedia parallel corpus (Wolk and Marasek, 2014), the Mozilla and EUbookshop parallel corpora (Tiedeman, 2012), and the bible parallel corpus (Christodoulopoulos and Steedman, 2015). We observe (see §8) a similar trend to the general case, which reaffirms our results. In fact, there seem to be bigger gains over the Transformer, albeit not drastically, compared to the general case.

A.4 Sign-Test

In order to assess the consistency of the improvements of our models, we perform the Sign-Test on every two models (see §5). Evidently, SASA persistently outperforms the Transformer baseline and the syntactic models, as does the combined model of SASA and UDISCAL.

A.5 SemSplit

Following Sulem et al. (2020), we implement the SemSplit pipeline. First, we train a Transformer-based Neural Machine Translation model. Then, during inference time, we use the Direct Semantic
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>En-De</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bleu</td>
<td>Transformer</td>
<td>17.6</td>
<td>20.55</td>
<td>22.17</td>
<td>25.46</td>
<td>19.7</td>
<td>28.01</td>
<td>26.84</td>
<td>17.71</td>
<td>16.94</td>
<td>21.66</td>
</tr>
<tr>
<td>chrF</td>
<td>Transformer</td>
<td>47.37</td>
<td>51.85</td>
<td>52.52</td>
<td>55.06</td>
<td>50.87</td>
<td>57.81</td>
<td>55.48</td>
<td>45.19</td>
<td>44.18</td>
<td>51.15</td>
</tr>
<tr>
<td></td>
<td>SemSplit</td>
<td>43.42</td>
<td>47.19</td>
<td>47.05</td>
<td>49.86</td>
<td>45.87</td>
<td>51.50</td>
<td>50.24</td>
<td>47.71</td>
<td>46.93</td>
<td>47.75</td>
</tr>
<tr>
<td><strong>En-Ru</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bleu</td>
<td>Transformer</td>
<td>24.32</td>
<td>18.11</td>
<td>25.35</td>
<td>21.1</td>
<td>19.77</td>
<td>22.34</td>
<td>19</td>
<td>20.14</td>
<td>15.64</td>
<td>22.33</td>
</tr>
<tr>
<td>chrF</td>
<td>Transformer</td>
<td>51.39</td>
<td>45.69</td>
<td>53.31</td>
<td>50.16</td>
<td>48.10</td>
<td>50.54</td>
<td>48.01</td>
<td>45.78</td>
<td>42.51</td>
<td>53.07</td>
</tr>
<tr>
<td></td>
<td>SemSplit</td>
<td>46.10</td>
<td>40.50</td>
<td>47.66</td>
<td>44.58</td>
<td>43.16</td>
<td>45.34</td>
<td>43.38</td>
<td>40.97</td>
<td>38.93</td>
<td>47.84</td>
</tr>
<tr>
<td><strong>En-Fi</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bleu</td>
<td>Transformer</td>
<td>11.22</td>
<td>12.76</td>
<td>10.2</td>
<td>13.35</td>
<td>11.37</td>
<td>9.32</td>
<td>12.21</td>
<td>11.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SemSplit</td>
<td>6.97</td>
<td>7.72</td>
<td>6.55</td>
<td>8.75</td>
<td>7.54</td>
<td>6.18</td>
<td>7.73</td>
<td>7.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chrF</td>
<td>Transformer</td>
<td>43.79</td>
<td>45.48</td>
<td>43.43</td>
<td>46.39</td>
<td>43.96</td>
<td>42.06</td>
<td>43.10</td>
<td>44.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SemSplit</td>
<td>40.18</td>
<td>41.42</td>
<td>39.94</td>
<td>42.18</td>
<td>40.20</td>
<td>38.76</td>
<td>40.12</td>
<td>40.40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **En-Tr** |       |      |      |      |      |      |      |      |      |       |         |
| Bleu   | Transformer | 8.43 | 8.55 | 8.1 | 8.36 |       |       |       |       |       |         |
|        | SemSplit   | 6.15 | 6.07 | 5.37 | 5.86 |       |       |       |       |       |         |
| chrF   | Transformer | 40.24 | 40.37 | 39.75 | 40.12 |       |       |       |       |       |         |
|        | SemSplit   | 39.04 | 39.00 | 38.85 | 38.97 |       |       |       |       |       |         |

Table 6: Bleu and ChrF scores of the baseline Transformer and the SemSplit model.
Table 7: ChrF scores for the baseline Transformer model, the baseline Syntactically infused models PASCAL and UDICsAL, our SASA and SACrA models, and models incorporating UDICsAL with each of SASA and SACrA, across all WMT’s newstests. For every language pair, each column contains the Bleu scores over the WMT newstest equivalent to the column’s year (e.g., for En-Ru, the scores under column 2015 are for En-Ru newstest2015). For some newstests, there was more than one version on WMT, each translated by a different person. For those test sets, we included both versions, denoting the second one with a “B”. In addition, for every language pair, the right-most column represents the average Bleu scores over all the pair’s reported newstests. For every test set (and for the average score), the best score is boldfaced. For each of the semantic models (i.e., SASA and SACrA), improvements over all the baselines (syntactic and Transformer) are marked by an arrow facing upwards. For models with both syntactic and semantic masks, improvements over each mask individually are marked by an asterisk.
Table 8: Bleu scores of challenge sentences for the baseline Transformer model, the baseline Syntactically infused models PASCAL and UDISCAL, our SASA and SACrA models, and models incorporating UD ISC AL with each of SASA and SACrA, across all WMT’s newstests. For every language pair, each column contains the Bleu scores over the WMT newstest equivalent to the column’s year (e.g., for En-Ru, the scores under column 2015 are for En-Ru newstest2015). For some newstests, there was more than one version on WMT, each translated by a different person. For those test sets, we included both versions, denoting the second one with a “B”. In addition, for every language pair, the right-most column represents the average Bleu scores over all the pair’s reported newstests. For every test set (and for the average score), the best score is boldfaced. For each of the semantic models (i.e., SASA and SACrA), improvements over all the baselines (syntactic and Transformer) are marked by an arrow facing upwards. For models with both syntactic and semantic masks, improvements over each mask individually are marked by an asterisk.