AMR Alignment: Paying Attention to Cross-Attention

Anonymous ACL submission

Abstract

With the surge of Transformer models, many have investigated how attention acts on the learned representations. However, attention is still overlooked for specific tasks, such as Semantic Parsing. A popular approach to the formal representation of a sentence's meaning is Abstract Meaning Representation (AMR). Until now, the alignment between a sentence and its AMR representation has been explored in different ways, such as through rules or via the Expectation Maximization (EM) algorithm. In this paper, we investigate the ability of Transformer-based parsing models to yield effective alignments without ad-hoc strategies. We present the first in-depth exploration of cross-attention for AMR by proxy of alignment between the sentence spans and the semantic units in the graph. We show how current Transformer-based parsers implicitly encode the alignment information in the cross-attention weights and how to leverage it to extract such alignment. Furthermore, we supervise and guide cross-attention using alignment, dropping the need for English- and AMR-specific rules.

1 Introduction

At the core of NLU lies the task of Semantic Parsing, aiming at translating natural language text into machine-interpretable representations. One of the most popular semantic formalisms is the Abstract Meaning Representation (Banarescu et al., 2013, AMR), which embeds the semantics of a sentence in a directed acyclic graph, like shown in Figure 1, where concepts are represented with nodes, such as *thirst*; semantic relation between concepts are represented by edges, such as *:purpose*; and the co-references are represented with reentrant nodes, such as *p4* representing *pill*. As of now, AMR has been widely used in Machine Translation (Song et al., 2019), Question Answering (Lim et al., 2020; Bonial et al., 2020b; Kapanipathi et al., 2021), Human-Robot Interaction (Bonial et al., 2020a), Text Summarization (Hardy and Vlachos, 2018; Liao et al., 2018) and Information Extraction (Rao et al., 2017), among other areas.

Alignment between spans in text and semantic units in graphs (see Figure 1) is a fundamental requirement for multiple purposes, such as training AMR parsers (Wang et al., 2015; Flanigan et al., 2016; Misra and Artzi, 2016; Damonte et al., 2017; Zhou et al., 2021), cross-lingual AMR parsing (Blloshmi et al., 2020), applying AMR in downstream tasks (Song et al., 2019), or the creation of new semantic parsing formalisms (Navigli et al., 2022; Martínez Lorenzo et al., 2022), among others. However, AMR does not provide such alignment information.

Several alignment standards have been proposed to mitigate this issue, such as JAMR (Flanigan et al., 2014), ISI (Pourdamghani et al., 2014) or LEAMR (Blodgett and Schneider, 2021), among others. Following these standards, there are different approaches to extract the alignments, such as adopting rule-based approaches (Liu et al., 2018), or by statistical strategies using Expectation Maximization (EM) (Pourdamghani et al., 2014; Blodgett and Schneider, 2021).

Current state-of-the-art AMR parsers are auto-regressive neural models (Bevilacqua et al., 2021; Bai et al., 2022) that do not generate or rely on alignment when parsing the sentence to produce the graph. Therefore to obtain both, one needs to i) predict the graph and then ii) generate the alignment using an aligner system.

Recent work has questioned considering attention as an explanation (Bibal et al., 2022), or put it to test against other approaches such as saliency methods (Bastings and Filippova, 2020). We want to explore whether this holds true for cross-attention in auto-regressive parsers and the alignment problem in Semantic Parsing as we uncover the relation between them. This paper ex-

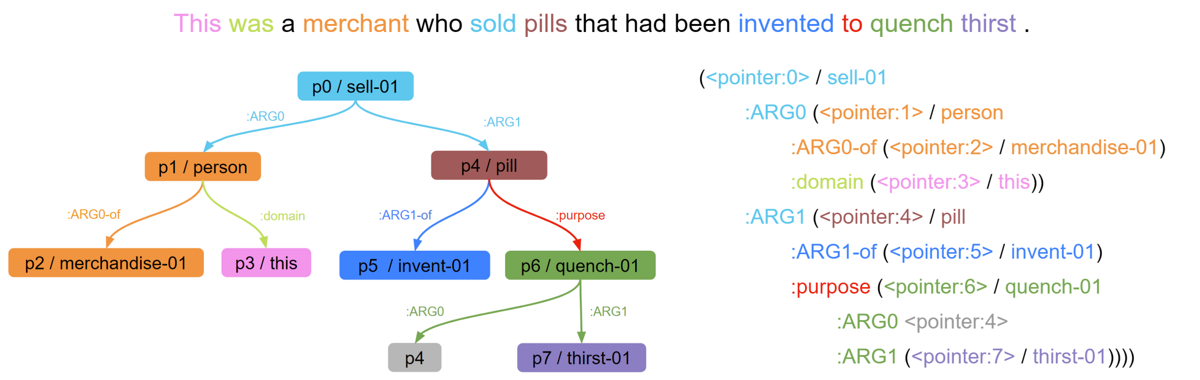

Figure 1: A sentence (top) with its AMR graph (left) and the AMR linearization (right). Colors represent alignment.

plores how auto-regressive models implicitly encode the relations between spans in the text and semantic units in the graph through cross-attention and how the alignment can be obtained directly while predicting the graphs.

The main contributions of this paper are as follows: (i) we explore the type of implicit alignment knowledge that Transformer-based AMR parsing models preserve; (ii) we extract the alignment information from the model; (iii) we supervise a model's cross-attention for improving how it learns and (iv) obtain state-of-the-art results in AMR alignment, along with different standards.

2 Related Work

JAMR (Flanigan et al., 2014) aligns spans to subgraphs by applying an ordered list of 14 criteria. One shortcoming of JAMR is that it is unable to resolve ambiguities. **TAMR** (Liu et al., 2018) extends it with an oracle parser that selects the alignment corresponding to the highest-scored candidate AMR graph. **ISI** (Pourdamghani et al., 2014) uses an EM algorithm to establish alignments of tokens with nodes and relations. The graph is first linearized, and then EM is used with a symmetrized scoring function where probabilities of a node or edge to be aligned to a word and vice versa are equal.

This leads to more diversity in terms of alignment patterns but fails when facing easy to recognize patterns such as dates.

LEAMR (Blodgett and Schneider, 2021) combines rules and EM to automatically align sentence spans with graph's semantic units. All semantic units in the graph should be aligned to at least one span of the sentence, which makes it the first standard to tackle reentrant nodes.

Throughout the last years several systems have incorporated innovative methods to extract the alignment, e.g., by incorporating syntactic information (Chen and Palmer, 2017; Szubert et al., 2018; Chu and Kurohashi, 2016), word embeddings (Anchiêta and Pardo, 2020) or including graph distance information (Wang and Xue, 2017). Zhou et al. (2021) provide alignments while parsing with a transition based approach, but rely on JAMR alignments and are not evaluated.

2.1 Semantic Parsing and Transformer

Most modern systems for AMR parsing rely on Encoder-Decoder Transformers such a BART or T5 (Lewis et al., 2020; Raffel et al., 2020; Lam et al., 2021). Such models consist of two stacks of Transformer layers, with self- and cross-attention as their backbone.

With the surge of Transformer models, research has explored how attention encodes the information in text, i.e., whether it corresponds to the intuition behind human attention (Vashishth et al., 2019), or different definitions of explainability (Bastings and Filippova, 2020; Bibal et al., 2022). Several works have investigated how attention operates, relates to preconceived ideas, aggregates information and explains model behavior for tasks such as Natural Language Inference (Stacey et al., 2021), Translation (Yin et al., 2021; Zhang and Feng, 2021; Chen et al., 2021), Summarization (Xu et al., 2020; Manakul and Gales, 2021) or Sentiment Analysis (Wu et al., 2020). There have even been attempts at guiding attention in order to improve interpretability or its performance in downstreams tasks (Deshpande and Narasimhan, 2020; Sood et al., 2020). However, to our knowledge there has been no study on attention for AMR Parsing. We fill this gap with our paper.

3 Foundations

3.1 Alignment Standards

While conceptually our approach is agnostic to different standards, we rely on existing ones. Figure 1 shows an intuition of the concept's alignment between the sentence and the AMR graph.

ISI The ISI standard aligns single spans in the sentence to graphs' semantic units (nodes or relations). ISI aligns relations and reentrant nodes when they explicitly appear in the sentence.

LEAMR The LEAMR standard differentiates among 4 different types of alignment: i) Subgraph Alignments, where all the subgraphs that explicitly appear in the sentence are aligned to a list of consecutive spans, ii) Duplicate Subgraph, where all the subgraphs that represent omitted concepts in the sentence are aligned, iii) Relation Alignments, where all the relations that do not take part in a previous subgraph structure are aligned, and iv) Reentrancy Alignments, where all the reentrant nodes are aligned. In contrast to ISI, all the semantic units in the graph are aligned to some list of consecutive spans in the text.

3.2 Cross-attention

Originally described by Vaswani et al. (2017) as "multi-head attention over the output of the encoder", and referred to as **cross-attention** in Lewis et al. (2020); it enables the Decoder to attend to the output of the Encoder stack, conditioning the hidden states of the autoregressive component on the input text. We define the self-attention module and Transformer cross-attention as:

$$\text{Attention}(Q, K, V) = \text{att}(Q, K)V$$

$$\text{att}(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}})$$

$$\text{CrossAtt}(Q, K, V) =$$

$$\text{Concat}(head_1, ..., head_H)W^O$$

$$head_h = \text{Attention}(QW_h^Q, KW_h^K, VW_h^V)$$

where $K, V = E^\ell \in \mathbb{R}^{n_e \times d_k H}$ and $Q = D^\ell \in \mathbb{R}^{n_d \times d_k H}$ are the encoder and decoder hidden states at layer ℓ, n_e and n_d are the input and output sequence lengths, H the number of heads, W_h^Q, W_h^K and $W_h^V \in \mathbb{R}^{d_k H \times d_k}$ are learned weights that project the hidden states to the appropriate dimensions, d_k, for each head and $W^O \in \mathbb{R}^{d_k H \times d_k H}$ is a final learned linear projection. Therefore in each head h and layer ℓ we define the attention weights as $att_h^\ell = \text{att}(D^\ell W_h^Q, E^\ell W_h^K) \in \mathbb{R}^{n_d \times n_e}$.

4 Method

4.1 Unsupervised Cross-Attention

We argue there is an intuitive connection between cross-attention and alignments. Under the assumption the decoder will attend to the parts in the input that are more relevant to predict the next token, we infer that when decoding the tokens for a certain node in the graph, attention should focus on related tokens in the input, and therefore the words that align to that node. We will use the cross-attention matrices (att_h^ℓ) to compute an alignment between the input and the output.

4.2 Guided Cross-Attention

We want to explore whether cross-attention can be guided by the alignment between the words of the sentence and nodes of the semantic graph. To this end, we construct a sparse matrix $align \in \mathbb{R}^{n_d \times n_e}$ from the automatically generated ISI or LEAMR alignments:

$$align(i, j) = \begin{cases} 1 & if \quad x_i \sim y_j \\ 0 & if \quad x_i \nsim y_j \end{cases}$$

where $\sim$ indicates an alignment between token x_i (part of a word) and graph-token y_j (part of a node or relation).

However this produces a sparse matrix. While there are sparse versions of attention (Martins and Astudillo, 2016), they did not produce successful alignments in our experiments. Hence we choose to alleviate the constraint of imposing sparsity by employing the scalar mixing approach introduced in ELMO (Peters et al., 2018). We therefore learn a weighted mix of each head and obtain a single attention matrix:

$$att^\ell = \gamma \sum_{h=0}^{H} s_h att_h^\ell \in \mathbb{R}^{n_d \times n_e} \tag{1}$$

where $\mathbf{s} = softmax(\mathbf{a})$ with scalar learnable parameters $\gamma, a_0, \ldots, a_H$. We obtain better results when using a subset of heads to compute att^ℓ.

The model is free to give more weight to certain heads that naturally become more sparse, while other heads are free to encode useful information that may be independent from alignment. In our experiments we use the implementation of Bevilacqua et al. (2021) to train our parser, but add an additional Cross-Entropy loss signal:

$$\mathcal{L} = \mathcal{L}_{LM} - \sum_{\substack{j \\ \sum_i align(i,j)>0}}^{n_d} \sum_{i}^{n_e} \log \frac{\exp\left(att^{\ell}(i,j)\right)}{\sum_k^{n_d} \exp\left(att^{\ell}(i,k)\right)}$$

$$\frac{align(i,j)}{\sum_k^{n_e} align(k,j)}$$

4.3 Saliency Methods

Input saliency methods represent a theoretically-valid alternative to our reasoning about cross-attention, i.e. that when decoding the tokens for a certain node in the graph, a higher importance will be given to the tokens in the input that correspond to that node, or at least to those that were more important in their prediction.

Therefore we look at the saliency weights of the input at each decoding step, obtaining a weight matrix with the same size as the cross-attention, $sal \in \mathbb{R}^{n_d \times n_e}$.

To this end we deploy Captum (Kokhlikyan et al., 2020), with an array of saliency methods such as gradient-based: Integrated Gradients (IG), Saliency (Simonyan et al., 2014), Input X Gradient (IxG); backpropagation-based: Deeplift (Shrikumar et al., 2017), Guided Backpropagation (GB) (Springenberg et al., 2015); and finally occlusion-based (Zeiler and Fergus, 2014).

5 Alignment Extraction

The algorithm [1] to extract and to align the input-output spans is divided into the following steps:

1. **Alignment Score Matrix** We create a matrix $M \in \mathbb{R}^{n_d \times n_e}$, where n_e is number of tokens in the sentence and n_d is the number of tokens in the linearized graph, using the cross-attention or saliency weights as described in Section 4.

2. **Span Segmentation** We sum the scores of tokens that belong to the same sentence words column-wise in M. Then, the sentence tokens are grouped into spans using the span segmentation procedure in LEAMR (Blodgett and Schneider, 2021).

3. **Graph Segmentation** We sum the score of tokens that belong to the same graph's semantic unit row-wise in M.

4. **Sentence Graph Tokens Map** We iterate over all the graph's semantic units and map them to the sentence span with higher score in M.

5. **Special Graph Structures** We revise the mapping by identify subgraphs that represent literal or matching spans – e.g., named entities, date entities, specific predicates, etc – and align them accordingly.

6. **Alignment Formatting** We extract the final alignments to the appropriate format using the resulting mapping relating graph's semantic units to sentence spans.

[1]The pseudo algorithm is described in the Appendix C

6 Experimental Setup

6.1 Datasets

Graph inventory AMR 3.0 (LDC2020T02) contains 59,255 manually annotated sentence-graph pairs. We only use the train split for the guided approach, and use the respective validation and test splits from the alignment systems.

Alignments We evaluate our systems on two gold alignments. **ISI** (Pourdamghani et al., 2014) released two splits of 200 manually annotated alignments that we use as validation and test set. We update them to the AMR 3.0 formalism. Similarly **LEAMR** provided 150 validation and 200 test manually annotated alignments. These include some sentence-graph pairs from *The Little Prince* Corpus (TLP) complemented with randomly sampled from AMR 3.0.

6.2 Model

In all cases we use **SPRING** as our parsing model, based on BART-large. We extract all att_h^{ℓ} matrices from a model trained on AMR 3.0 as in Blloshmi et al. (2021) in order to perform our unsupervised cross-attention analysis. For the guided approach we re-train using the same hyper-parameters as the original implementation but with an extra loss signal as described in Section 4.2 based on either LEAMR or ISI. When using LEAMR alignments, we restructure the training split in order to exclude any pair from their test and validation sets.

7 Experiments

Layer and Head analysis To explore how cross-attention correlates to alignment, we compute the Pearson's r correlation between each att_h^{ℓ} matrix and the LEAMR alignment matrix $align$ after we

Figure 2: Unsupervised (left), saliency (center-left) and guided (center-right) alignment weights and LEAMR (right) gold alignment for lpp_1943.1209. To interactively explore all cross-attention weights go here.

flatten them and remove special tokens not relevant for alignment. In Figure 3 we observe how, overall, there is a clear positive correlation. We noticed that attention is focused solely on the beginning and end of sentence tokens and punctuation marks in heads with a low correlation. While we do not have an intuition on why certain heads correlate more with it, there is a clear connection between cross-attention and alignment. For instance, the head 6 in layer 3 (att^3_6) achieves a value of 0.635, approximately the same as the sum of the whole layer. The left image in Figure 2 shows the cross-attention values for att^3_6 for an example of the TLP corpus. Notice how despite being a model that has not seen any alignment information, it can find the correct correspondence between non-trivial matches such as $merchant$ and $person$.

Saliency methods The two most correlated methods were Saliency and GB, with 0.575. Despite this result, when we look at it, we notice how saliency methods were more prone to focus on essential parts of the sentence, such as the subject or predicate. These are usually aligned to more nodes and relations, explaining the high correlation, but it was less nuanced than cross-attention. The center image of Figure 2 portrays such conduct.

Guided Our best result was by supervising layer 3 during training using the approach described in 4.2, on half of the heads (3, 4, 5, 6, 7, 11, 12 and 15) selected by their correlation on the validation set and using Cross-Entropy Loss. The performance on parsing was not affecte, there is more information in Appendix D. When we look at att^3 using the learned weighted mix from Equation 1 with LEAMR alignments, the correlation reaches 0.866, much higher than any other method. Figure 3 shows the impact of supervising half the heads on layer 3, as well as how it even influences heads in other layers. By looking at the center-right Figure 2, att^3_6 attention is more condensed, which ties with the improvement in correlation. However, notice how sometimes the model confidently attends to incorrect positions, such as <$pointer$:0> and $merchant$ when it should be $sold$.

8 Results

Table 1 shows the performances of our two approaches on the LEAMR gold alignments compared to previous systems. We use the same evaluation setup as Blodgett and Schneider (2021), where the partial match assigns a partial credit from Jaccard indices between nodes and tokens. In both guided and unsupervised methods, we extract the score matrix for Algorithm 0 from the sum of the cross-attention in the first four layers. We used a Wilcoxon signed-rank test (Wilcoxon, 1945) on the alignment matches per graph to check for significant differences. Both our approaches were significantly different compared to LEAMR (p=0.031 and p=0.007 respectively). However, we found no statistical difference between our unsupervised and guided approaches (p=0.481).

Our guided attention approach performs best, improving upon LEAMR on Subgraph (+0.5) and Relation (+2.6). For Reentrancy, performance is relatively low, and we will explore the reasons behind

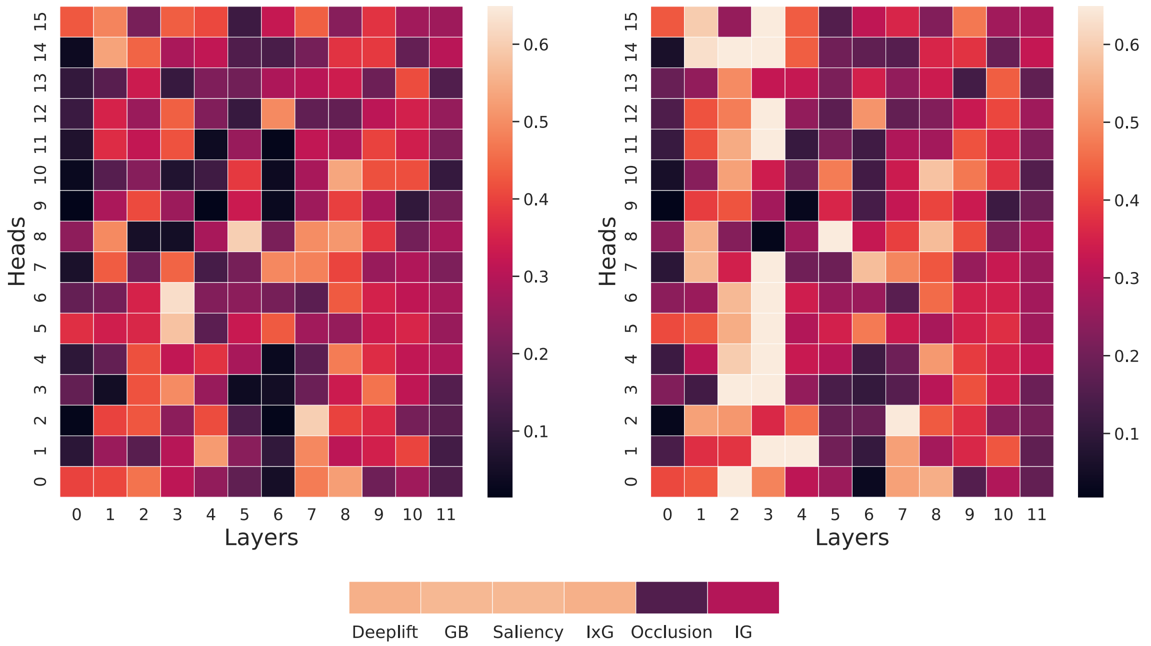

Figure 3: Heatmap of Pearson's R correlation to LEAMR validation set for unsupervised (left) and guided (right) cross-attention weights as well as saliency methods (bottom).

such scores later. Perhaps most interesting is the performance of the unsupervised alignment system using raw cross-attention weights from SPRING. It stays competitive against the guided model without having access to any alignment information. It outperforms LEAMR which, despite being unsupervised, relies on a set of inductive biases and rules based on alignments. While we also draw on specific rules related to the graph structure in post-processing, we will investigate their impact in an ablation study.

Relations that are argument structures (i.e.:ARG and :$ARG\text{-}of$) usually depend on the predictions for their parent or child nodes; hence their improvement is tied to the Subgraph Alignment. The results in Table 2 reassure this intuition. Notice how for Single Relations (such as :*domain* or :*purpose* in Figure 2) the performance by LEAMR was much lower, even worse than that of ISI. Blodgett and Schneider (2021) argued that it was due to the model being overeager to align to frequent prepositions such as *to* and *of*. On the other hand, our unsupervised method achieves 15 points over ISI and 20 over LEAMR, which hints at the implicit knowledge on alignment that cross-attention encodes. Our guided approach experiences a considerable drop for Single Relations since it was trained on data generated by LEAMR, replicating its faulty behavior albeit being slightly more robust.

When we test our systems against the ISI alignments instead, both our models achieve state-of-the-art results, surpassing those of previous systems, including LEAMR. This highlights the flexibility of cross-attention as an standard-agnostic aligner. We provide additional information in Appendix B.

9 Ablation

To get further insights on the results we perform an ablation study on:

Gold spans LEAMR relies on a span segmentation phase, with a set of multiword expressions as well as Stanza based named entity identification. We use the same system in order to have matching sentence spans, however these sometimes differ from those in the gold data leading to errors. Top of Table 3 shows the performance when the gold spans from the test set are used instead. We see how performance improves across all systems and our approach shows gains over LEAMR independent of correct spans.

Rules All modern alignment systems have a certain dependency on rules. For instance, we use the subgraph structure for Named Entities, certain relations are matched to their parent or child nodes, etc. See Appendix A for more details. But what is the impact of such rules? As expected, both LEAMR and our unsupervised method see a considerable performance drop. For Relation, LEAMR

		Exact Alignment			Partial Alignment			Spans	Coverage
		P	R	F1	P	R	F1	F1	
Subgraph Alignment (1707)	ISI	71.56	68.24	69.86	78.03	74.54	76.24	86.59	78.70
	JAMR	87.21	83.06	85.09	90.29	85.99	88.09	92.38	91.10
	TAMR	85.68	83.38	84.51	88.62	86.24	87.41	94.64	94.90
	LEAMR	93.91	94.02	93.97	95.69	95.81	95.75	96.05	100.00
	LEAMR †	93.74	93.91	93.82	95.51	95.68	95.60	95.54	100.00
	Ours - Unsupervised	94.11	94.49	94.30	96.03	96.42	96.26	95.94	100.00
	Ours - Guided - ISI	89.87	91.97	90.91	92.11	94.27	93.18	93.69	100.00
	Ours - Guided - LEAMR	**94.39**	**94.67**	**94.53**	**96.62**	**96.90**	**96.76**	96.40	100.00
Relation Alignment (1263)	ISI	59.28	8.51	14.89	66.32	9.52	16.65	83.09	9.80
	LEAMR	85.67	87.37	85.52	88.74	88.44	88.59	95.41	100.00
	LEAMR †	84.63	84.85	84.74	87.77	87.99	87.88	91.98	100.00
	Ours - Unsupervised	87.14	87.59	87.36	89.87	90.33	90.10	91.03	100.00
	Ours - Guided - ISI	83.82	83.39	83.61	86.45	86.00	86.22	87.30	100.00
	Ours - Guided - LEAMR	**88.03**	**88.18**	**88.11**	**91.08**	**91.24**	**91.16**	91.87	100.00
Reentrancy Alignment (293)	LEAMR	55.75	54.61	55.17					100.00
	LEAMR †	54.61	54.05	54.33					100.00
	Ours - Unsupervised	44.75	44.59	44.67					100.00
	Ours - Guided - ISI	42.09	39.35	40.77					100.00
	Ours - Guided - LEAMR	**56.90**	**57.09**	**57.00**					100.00
Duplicate Subgraph Alignment (17)	LEAMR	66.67	58.82	62.50	70.00	61.76	65.62		100.00
	LEAMR †	68.75	64.71	66.67	68.75	64.71	66.67		100.00
	Ours - Unsupervised	**77.78**	**82.35**	**80.00**	**77.78**	**82.35**	**80.00**		100.00
	Ours - Guided - ISI	63.16	70.59	66.67	65.79	73.53	69.44		100.00
	Ours - Guided - LEAMR	70.00	82.35	75.68	72.50	85.29	78.38		100.00

Table 1: LEAMR alignments results. Column blocks: models; Exact and Partial scores; Span and Coverage measures. Row blocks: alignment types, number of instances in brackets. † indicates our re-implementation. Guided versions using ISI/LEAMR silver alignments. Bold is best.

	AMR parser	P	R	F1
ALL	ISI	59.3	08.5	14.9
	LEARM †	84.6	84.9	84.7
	Ours - Unsupervised	87.1	87.6	87.4
	Ous - Guided - LEAMR	**88.0**	**88.2**	**88.1**
Single Relations (121)	ISI	**82.9**	52.1	64.0
	LEARM †	64.8	55.7	59.9
	Ours - Unsupervised	79.5	**79.5**	**79.5**
	Ous - Guided - LEAMR	77.5	64.8	70.5
Argument Structure (1042)	ISI	39.6	03.5	06.4
	LEARM †	86.6	88.2	87.4
	Ours - Unsupervised	87.9	88.4	88.2
	Ous - Guided - LEAMR	**89.0**	**90.8**	**89.9**

Table 2: LEAMR results breakdown for Relation Alignment. Column blocks: relation type; models; scores. † indicates our re-implementation. Bold is best.

drops by almost 60 points, since it heavily relies on the predictions of parent and child nodes to provide candidates to the EM model. Our unsupervised approach also suffers from such dependency, losing 25 points. However, our guided model is quite resilient to rules removal, barely dropping by one point on Subgraph and 5 on Relation.

Layers Figure 3 showed how alignment acts differently across heads and layers. We explore this information flow in the decoder by extracting the alignments from the sum of layers at different depths. The bottom of Table 3 shows this for both our unsupervised and guided models, as well as the Saliency method. [3] indicates the sum of heads in the supervised layer, while [3]* is the learned weighted mix. From our results early layers seem to align more explicitly, with performance dropping with depth. This corresponds to the idea that Transformer models encode basic semantic information early (Tenney et al., 2019). While layers 7 and 8 did show high correlation values, the cross-attention becomes more disperse with depth, probably due to each token encoding more contextual information.

	GOLD			Without Rules			Layers										
							Unsupervised					Guided					
	LEAMR †	Uns.	Guided	LEAMR †	Uns.	Guided	Sal.	[0:4]	[4:8]	[8:12]	[0:12]	[0:4]	[4:8]	[8:12]	[0:12]	[3]	[3]*
Sub.	96.5	96.7	**97.0**	87.6	88.6	**93.4**	62.2	**94.3**	69.8	63.3	87.7	**94.5**	74.4	66.3	93.2	93.7	93.7
Rel.	87.1	89.2	**90.3**	26.6	60.1	**83.4**	50.0	**87.7**	72.7	61.6	84.5	**88.1**	73.8	62.5	87.9	86.2	85.9
Reen.	56.8	46.7	**59.0**	15.2	38.6	**57.0**	34.5	**44.7**	41.1	36.1	41.9	**57.0**	39.2	33.0	51.0	52.7	53.4
Dupl.	62.9	**80.0**	75.7	40.0	71.8	**73.7**	9.5	**80.0**	11.1	27.3	64.3	**75.9**	30.0	27.3	66.7	70.3	70.3

Table 3: F1 results on Exact Alignment on ablation studies. Column blocks: alignment types; using gold spans; removing rules from the models; by layers. Guided approach using LEAMR silver alignments. † indicates our re-implementation. [x:y] indicates sum from layer x to y. * indicates weighted head sum. Bold is best.

	P	R	F1
JAMR	92.7	80.1	85.9
TAMR	92.1	84.5	88.1
LEAMR	85.9	92.3	89.0
Ours - Unsupervised	95.4	93.2	94.3
Ours - Guided	**96.3**	**94.2**	**95.2**

Table 4: ISI results. Column blocks: models, measures.

ISI Table 4 shows the performance of our systems and previous ones with the ISI alignment as reference. We omitted relations and Named Entities in order to focus solely on non-rule based alignments and have a fair comparison between systems. Here, our aligner does not rely on any span-segmentation, hence nodes and spans are aligned solely based on which words and nodes share the highest cross-attention values. Still, Over the previous systems, ours outperformed by over 5 points

10 Error analysis

We identify three main classes of errors that undermine the extraction of alignments:

Consecutive spans Because each subgraph in LEAMR is aligned to a list of successive spans, the standard cannot correctly deal with transitive phrasal verbs. For example, for the verb "take off" the direct object might appear in-between ("take your jacket off"). Because these are not consecutive spans, we align just to "take" or "off".

Rules We have a few rules to recognize subgraph structures, such as Named Entities, and align them to the same spans. However, Named Entity structures contain a placeholder node indicating the entity type and when the placeholder node appears explicitly in the sentence, the node should not be part of the Named Entity subgraph. For example, when aligning *'Málaga', the city*, the placeholder node should be aligned to *city* while our model aligned it to *Málaga*.

Reentrancy Because all graph units in LEAMR must be aligned, Reentrancy performs the poorest compared to the other types. For example, in the sentence *He wants to protect himself* the primary node is *He* and there are two reentrant nodes, one referring to who protects – this is omitted in the sentence – and the other one to who is protected (*himself*). The LEAMR standard aligns the non-omitted nodes to the sentence's specific word that reflects the meaning (*himself*) and the omitted to the main verb (*protect*). However, the unsupervised model fails to align the reentrant nodes that is omitted in the sentence. On the other hand the guided model sometimes fails to align the node that appears explicitly in the sentence correctly. We blame this to the silver nature of the train data, which propagates the LEAMR error which usually just aligned these words to the verb.

11 Conclusion and Implications

In this paper we show for the first time how cross-attention is closely tied to the concept of alignment for Semantic Parsing in AMR. Both our unsupervised and our guided attention systems outperform previous alignment models. Moreover, our proposed method uses the cross-attention from a state-of-the-art parsing model, with no overhead computation and without influence the performance in the parsing task. The fact that our approach is much more resilient to the lack of handcrafted rules shows its capability as a standard-agnostic aligner, opening the door to its use in other tasks such as Machine Translation or Summarization.

In the future, with the objective of obtaining the first language-agnostic AMR aligner system, we aim to explore its zero-shot capabilities on cross-lingual AMR parsing. Furthermore, we are interested in perform an analysis about what are attending the attention heads that are not correlate to the alignment information.

12 Ethical Considerations

Regarding the ethical and social implications of our approach for AMR alignments, we do not believe it could have a negative impact. However methods such as guiding cross-attention could introduce new ways to supervise a model in order to produce harmful or unwanted model predictions.

References

Rafael Anchiêta and Thiago Pardo. 2020. Semantically inspired AMR alignment for the Portuguese language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1595–1600, Online. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Souza Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Association for Computational Linguistics. Annual Meeting. Conference Proceedings*.

Rexhina Blloshmi, Michele Bevilacqua, Edoardo Fabiano, Valentina Caruso, and Roberto Navigli. 2021. SPRING Goes Online: End-to-End AMR Parsing and Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 134–142, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2487–2500, Online. Association for Computational Linguistics.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3310–3321, Online. Association for Computational Linguistics.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020a. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss. 2020b. InfoForager: Leveraging semantic search with AMR for COVID-19 research. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 67–77, Barcelona Spain (online). Association for Computational Linguistics.

Chi Chen, Maosong Sun, and Yang Liu. 2021. Mask-align: Self-supervised neural word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4781–4791, Online. Association for Computational Linguistics.

Wei-Te Chen and Martha Palmer. 2017. Unsupervised AMR-dependency parse alignment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 558–567, Valencia, Spain. Association for Computational Linguistics.

Chenhui Chu and Sadao Kurohashi. 2016. Supervised syntax-based alignment between english sentences and abstract meaning representation graphs.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Ameet Deshpande and Karthik Narasimhan. 2020. Guiding attention for self-supervised learning with transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4676–4686, Online. Association for Computational Linguistics.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773, Brussels, Belgium. Association for Computational Linguistics.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Hoang Thanh Lam, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam M. Nguyen, Dzung T. Phan, Vanessa López, and Ramon Fernandez Astudillo. 2021. Ensembling graph predictions for amr parsing. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual.*

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using AMR for commonsense reasoning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2459–2471, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430, Brussels, Belgium. Association for Computational Linguistics.

Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041, Online. Association for Computational Linguistics.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

André F. T. Martins and Ramón F. Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1614–1623. JMLR.org.

Dipendra Kumar Misra and Yoav Artzi. 2016. Neural shift-reduce CCG semantic parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1775–1786, Austin, Texas. Association for Computational Linguistics.

Roberto Navigli, Rexhina Blloshmi, and Abelardo Carlos Martinez Lorenzo. 2022. BabelNet Meaning Representation: A Fully Semantic Formalism to Overcome Language Barriers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using Abstract Meaning Representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Ekta Sood, Simon Tannert, Philipp Mueller, and Andreas Bulling. 2020. Improving natural language processing tasks with human gaze-guided neural attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 6327–6341. Curran Associates, Inc.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2015. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2021. Supervising model attention with human explanations for robust natural language inference.

Ida Szubert, Adam Lopez, and Nathan Schneider. 2018. A structured syntax-semantics interface for English-AMR alignment. 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 1169–1180, New Orleans, Louisiana. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *CoRR*, abs/1909.11218.

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

Chuan Wang and Nianwen Xue. 2017. Getting the most out of AMR parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1268, Copenhagen, Denmark. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Zhengxuan Wu, Thanh-Son Nguyen, and Desmond Ong. 2020. Structured self-AttentionWeights encode semantics in sentiment analysis. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 255–264, Online. Association for Computational Linguistics.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.

Kayo Yin, Patrick Fernandes, Danish Pruthi, Aditi Chaudhary, André F. T. Martins, and Graham Neubig. 2021. Do context-aware translation models pay the right attention? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 788–801, Online. Association for Computational Linguistics.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Shaolei Zhang and Yang Feng. 2021. Modeling concentrated cross-attention for neural machine translation with Gaussian mixture model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1401–1411, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A LEAMR Alignment Rules

The LEAMR standard has some predefined strategies for alignments that were followed during their annotation, as well as fixed in their alignment pipeline along EM. We kept a few of them when extracting the alignment, just those related to the structure of the graph, and not to token matching between the sentence and the graph.

A.1 Subgraph

- Nodes *have-org-role-91* and *have-rel-role-91* follow a fixed structure related to a person ie. the sentence word *enemy* is represented as *person* → *have-rel-role-91* → *enemy*, therefore for such subgraphs we use the alignment from the child node.

- Similarly for Named Entities, we align the whole subgraph structure based on its child nodes which indicate its surfaceform. However this leads to some errors as described in Section 10.

- We align node *amr-unknown* to the question mark if it appears in the sentence.

A.2 Relations

- For the relation *:condition* we align it to the word *if* when it appears in the sentence.

- *:purpose* is aligned with *to* when in the sentence.

- *:ARGX* relations are aligned to the same span as the parent node, while *:ARGX-of* to that of the child, since they share the alignment of the predicate they are connected to.

- For *:mod* and *:duration* we use the alignment from the child node.

- For *:domain* and *:opX* we use the alignment from the parent node.

B Extra Results

B.1 LEAMR Results

We explore the variance with different seeds when guiding cross-attention. Table 1 reports on a single seed selected at random. Table 5 shows the results for five different seeds as well as the average and standard deviation. We observe some variance, especially for those alignment types with fewer elements; however, average performance is always higher than previous approaches.

C Alignment Extraction Algorithm

Algorithm 0 shows the procedure for extracting the alignment between spans in the sentence and the semantic units in the graphs, using a matrix that weights encoder tokens with the decoder tokens

D AMR parsing

Since our guided approach was trained with a different loss than the SPRING model, it could influence the performance in the Semantic Parsing task. Therefore, we tested our model also in the AMR parsing task using the test set of AMR 2.0 and AMR 3.0. Table 6 shows the result, where we can observe how our model preserves the performance on parsing.

E Hardware

Experiments were performed using a single NVIDIA 3090 GPU with 64GB of RAM and Intel® Core™ i9-10900KF CPU.

Training the model took 13 hours, 30 min per training epoch while evaluating on the validation set took 20 min at the end of each one. We selected the best performing epoch based on the SMATCH metric on the validation set.

		Exact Alignment			Partial Alignment			Spans
		P	**R**	**F1**	**P**	**R**	**F1**	**F1**
Subgraph Alignment (1707)	Run 1	**94.39**	**94.67**	**94.53**	**96.62**	**96.90**	**96.76**	**96.40**
	Run 2	93.79	93.85	93.82	96.22	96.27	96.25	96.05
	Run 3	94.26	94.32	94.29	96.60	96.66	96.63	96.34
	Run 4	94.20	94.26	94.23	96.47	96.53	96.50	96.22
	Run 5	93.81	94.14	93.98	95.81	96.14	95.97	95.73
	Average	94.09	94.25	94.17	96.34	96.50	96.42	96.15
	Std	0.27	0.30	0.28	0.34	0.30	0.32	0.27
Relation Alignment (1263)	Run 1	88.03	88.18	88.11	91.08	91.24	91.16	91.87
	Run 2	87.90	88.36	88.13	90.71	91.18	90.95	91.87
	Run 3	**88.61**	**88.61**	**88.61**	**91.44**	**91.44**	**91.44**	**91.95**
	Run 4	88.39	**88.61**	88.50	91.02	91.25	91.14	91.66
	Run 5	88.59	88.44	88.52	91.24	91.08	91.16	91.86
	Average	88.30	88.44	88.37	91.10	91.24	91.17	91.84
	Std	0.32	0.18	0.28	0.27	0.13	0.17	0.05
Reentrancy Alignment (293)	Run 1	56.90	57.09	57.00				
	Run 2	56.23	56.42	56.32				
	Run 3	**57.24**	**57.43**	**57.34**				
	Run 4	55.56	55.74	55.65				
	Run 5	55.22	55.41	55.31				
	Average	56.23	56.42	56.32				
	Std	0.86	0.86	0.86				
Duplicate Subgraph Alignment (17)	Run 1	70.00	**82.35**	75.88	72.50	85.29	78.38	
	Run 2	65.00	76.47	70.27	67.50	79.41	72.97	
	Run 3	70.00	**82.35**	75.68	70.00	82.35	75.68	
	Run 4	**73.68**	**82.35**	**77.78**	**76.32**	**85.29**	**80.56**	
	Run 5	70.00	**82.35**	75.68	70.00	82.35	75.68	
	Average	69.74	81.17	75.06	71.26	82.94	76.65	
	Std	3.09	2.63	2.82	3.33	2.46	2.90	

Table 5: Results on the LEAMR alignment for 5 seeds on the guided approach. Column blocks: runs; measures. Row blocks: alignment types; average and standard deviation (std). Bold is best.

	AMR 2.0	AMR 3.0
SPRING	84.3	83.0
Ours - Guided - ISI	84.3	83.0
Ours - Guided - Leamr	84.3	83.0

Table 6: AMR parsing ResultsBold is best.

F Data

The AMR data used in this paper is licensed under the *LDC User Agreement for Non-Members* for LDC subscribers, which can be found here. The *The Little Prince* Corpus can be found here from the Information Science Institute of the University of Southern California.

G Limitations

Even though our method is an excellent alternative to the current AMR aligner system, which is standard and task-agnostic, we notice some drawbacks when moving to other autoregressive models or languages:

Model In this work, we studied how Cross Attention layers retain alignment information between input and output tokens in auto-regressive

Algorithm 1 Procedure for extracting the alignment between spans in the sentence and the semantic units in the graphs, using a matrix that weights encoder tokens with the decoder tokens.

```
function ExtractAlignments(encoderTokens, decoderTokens, scoreMatrix)
    alignmentMap ← dict()
    spansList ← Spans(encoderTokens)                    ▷ Extract sentence spans as in LEAMR
    spanPosMap ← tok2span(encoderTokens)                ▷ Map input tokens to spans
    graphPosMap ← tok2node(decoderTokens)               ▷ Map output tokens to graph unit
    CombineSubwordTokens(scoreMatrix)
    for decoderTokenPos, GraphUnit in graphPosMap do
        encoderTokensScores ← scoreMatrix[decoderTokenPos]
        maxScorePos ← argmax(encoderTokensScores)
        alignmentMap[GraphUnit] ← SelectSpan(spansList, maxScorePos)
    end for
    fixedMatches ← getFixedMatches(graphPosMap)         ▷ Look for rule based matches
    alignmentMap ← applyFixedMatches(alignmentMap, fixedMatches)
    alignments ← FormatAlignment(alignmentMap)
    return alignments
end function
```

models. In Section 7, we examined which layers in state-of-the-art AMR parser models based on BART-large best preserve this information. Unfortunately, we cannot guarantee that these layers are optimal for other auto-regressive models, and so on. As a result, an examination of cross-attention across multiple models should be required before developing the cross-lingual application of this approach.

Sentence Segmentation It is necessary to apply LEAMR's Spam Segmentation technique to produce the alignment in LEAMR format (Section 5). However, this segmentation method has several flaws: i) As stated in Section 10, this approach does not deal appropriately with phrasal verbs and consecutive segments; ii) the algorithm is English-specific; it is dependent on English grammar rules that we are unable to project to other languages. Therefore we cannot extract the LEAMR alignments in a cross-lingual AMR parsing because we lack a segmentation procedure. However, although LEAMR alignment has this constraint, ISI alignment does not require any initial sentence segmentation and may thus be utilized cross-lingually.